EVALUATING THE PERFORMANCE OF MACHINE-LEARNING TECHNIQUES FOR RECOGNIZING CONSTRUCTION MATERIALS IN DIGITAL IMAGES

A Thesis
Presented to
The Reading Committee

By:

Abbas Rashidi

In Partial Fulfillment
of the Requirements for the Degree
"Master of Science" in the
School of Electrical and Computer Engineering

Georgia Institute of Technology
August 2013

EVALUATING THE PERFORMANCE OF MACHINE-LEARNING TECHNIQUES FOR RECOGNIZING CONSTRUCTION MATERIALS IN DIGITAL IMAGES

Approved by:

Dr. David Citrin, Advisor
School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. Aaron Lanterman
School of Electrical and Computer Engineering
Georgia Institute of Technology

Dr. Baabak Ashuri
School of Building Construction
Georgia Institute of Technology

ACKNOWLEDGEMENTS

First and foremost, I want to express my deepest sense of gratitude to my adviser, Prof. David Citrin. I also thank Prof. Aaron Lanterman and Prof. Baabak Ashuri, for serving on my thesis committee and for their critical review of this research work.

TABLE OF CONTENTS

LIST OF TABLES

Page

LIST OF FIGURES

LIST OF SYMBOLS AND ABBREVIATIONS

FPR	False Positive Rate
FNR	False Negative Rate
MLP	Multilayer Perceptron
RBF	Radial Basis Function
SVM	Support Vector Machine

SUMMARY

Digital images acquired at construction sites contain valuable information which is useful for different applications including As-built documentation of building elements, effective progress monitoring, structural damage assessment, and quality control of construction material. As a result, there is an increasing need for effective methods to recognize different building materials in digital images and videos.

Pattern recognition is a mature field of research within the area of image processing; however, its application in the area of civil engineering and building construction is only recent.

In order to develop any robust image recognition method, it is necessary to choose the optimal machine learning algorithm. This study focuses on evaluating the effect of choosing different machine learning algorithms on the performance of material's image recognition algorithms. To generate a robust color model for building material detection in an outdoor construction environment, a comparative analysis of three generative and discriminative machine learning algorithms, namely, multilayer perceptron (MLP), radial basis function (RBF), and support vector machines (SVMs), is conducted. The study is narrowed down into three classes of major building materials: concrete, plywood, and brick.

For training purposes a large-size data set including hundreds of images is collected. The comparison study is conducted by implementing necessary algorithms in MATLAB and testing over several construction-site images. To assess the performance of each technique, the results are compared with a manual classification of building

materials. In order to better evaluate the performance of each technique, experiments are conducted by taking pictures under various realistic jobsite conditions, e.g., different ranges of image resolutions, different distance of camera from object, and different types of cameras. Results indicate that material detection algorithms are able to robustly detect uniform material, i.e. bricks. In addition, the study illustrates that for all three categories of construction material, SVM outperforms the other two methods.

CHAPTER 1

INTRODUCTION

During the last decade, the architecture, engineering, construction, and facilities management (AEC/FM) industry has been focusing on the development of different aspects of information technologies to improve productivity [37], work efficiency, and automating processes in construction jobsites [18]. One class of these technologies is machine-learning techniques. Over the last few years, the use of machine-learning techniques has increased in many areas of civil engineering. These applications include geotechnical engineering, transportation engineering, structural analysis and design [17], structural damage assessment [14], structural dynamics, and control [15]. The application of machine-learning techniques in construction management is quite recent. These applications cover a very wide area of construction issues. Machine-learning techniques have been developed internationally to assist the managers or contractors in many crucial construction decisions. Some of these models were designed for cost estimation, decision making, predicting the percentage of mark up, predicting production rate, etc. [12].This research presents a comparative assessment of the performance of three machine-learning techniques for recognizing construction materials in digital images. Digital images acquired at construction site contain considerable up-do-date construction-project information that, if extracted, would be of great value for different applications including progress-monitoring by means of documentation, reporting, and communication. Therefore, the need to automatically process and extract the desired and relevant information from digital images has been growing rapidly. The primary research objectives of this work are the detection of materials such as concrete, brick, and plywood using image processing and comparison between various types of machine-learning recognizing construction materials.

In this research, the materials that used for recognizing construction are concrete, brick, and plywood materials. In the following section, a brief description for each category of these material is presented.

1.1 Construction Material

1.1.1 Concrete

Concrete is basically a mixture of two components: aggregates and paste (or binder). The paste is comprised of cement, supplementary cementitious materials and water. It binds the aggregates (sand and gravel or crushed stone) into a rocklike mass. The chemical reaction of the cementitious materials and water, called hydration, is the process by which paste hardens and binds the aggregates [1].

Aggregates are generally divided into two groups: fine and coarse. Fine aggregates consist of natural or manufactured sand with particle sizes ranging from dust-size up to 3/8 inch; coarse aggregates are those with particles ranging in size from 6 in. down to about 0.05 in. For pavement, it is common for the maximum aggregate size to range from 1 to 1.5 in. Aggregates are sized by passing them through screens with standard openings called sieves.

Selection of aggregates for use in concrete is important since they make up about 60% to 75% of the total volume of concrete. Aggregates should consist of particles with adequate strength and resistance to exposure conditions and should not contain materials that will cause a chemical reaction with the paste that may lead to deterioration of the concrete. Screening tests are available to determine the adequacy of aggregates for use in concrete.

The paste is composed of Portland cement, supplementary cementitious materials (fly ash), water, and entrapped air or purposely entrained air. Cement paste ordinarily constitutes about 25% to 40% of the total volume of concrete. The volume of cement is

usually between 7% and 15% and the water between 14% and 21%. Air content ranges up to about 8% of the volume of the concrete.

There are many different types of cement, including Portland cement and blended hydraulic cement. ASTM C150 and AASHTO M85 classify Portland cements by five chemical and compositional designations: Type I through Type V. ASTM C 595 and ASTM C 1157 classify blended hydraulic cements. Blended cements contain additional ingredients, such as fly ash, or blast furnace slag directly in the cement. Presently, most concrete for use in pavement is made with a Portland cement, but use of blended hydraulic cements appears to be increasing.

Concrete is a versatile and inexpensive material, with a vast range of applications around the home. Brick laying, constructing paths and driveways, and foundations to buildings and walls, are some of the practical applications. Concrete has a similarly wide and varied range in industrial applications. These include; bridge construction, motorways, curbs, walkways, and foundations to entire factories and industrial sites. Cut away of concrete showing aggregate and paste are shown in Fig. 1.1.

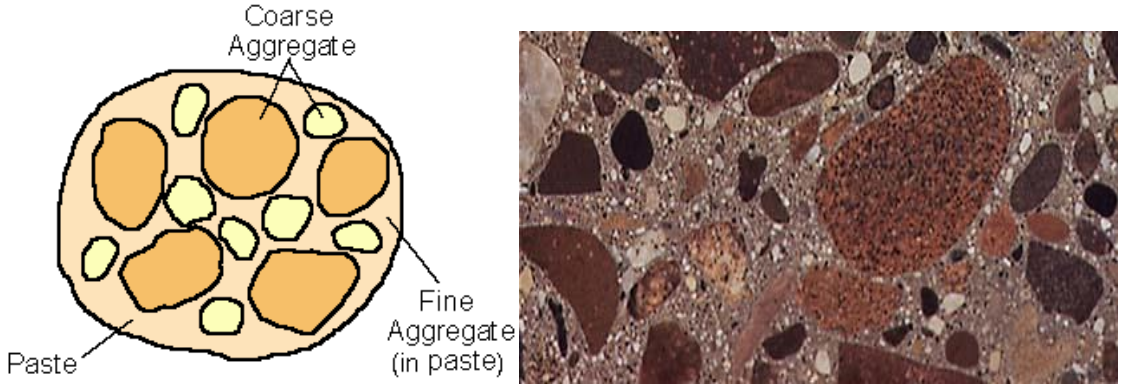


Fig. 1.1: Cut away of concrete showing aggregate and paste [1].

1.1.2 Brick material

Brick is the oldest manufactured building material, and much of its history is lost in antiquity. Bricks are used in a wide range of buildings from housing to factories, and in

the construction of tunnels, waterways, bridges etc. Their properties vary according to the purpose for which they are intended.

There are literally thousands of different bricks, but they can be broken down into a handful of basic types. The vast majority is made from clay and are kiln-fired. Some types of brick include the Wirecut, Stock, Handmade, Fletton, etc. Different visual textures of brick are shown in Fig. 1.2.

Wirecut **Stock** **Handmade** **Fletton**

Fig. 1.2: Different textures of brick[28].

Brick format

Modern metric bricks in the UK are sized to create a modular format. The standard brick size is 215 × 102.5 × 65mm (face × bed × end), so, with a standard 10mm wide joint, this gives a working size of 225mm × 75mm (Lester Woods General Builders). They are a contractor, but do they set standards?

Methods of manufacture

Modern clay bricks are formed in one of three processes including the soft mud, dry press, and extruded.

Normally, brick is contained the silica, alumina, lime, iron oxide, and magnesia**.**

1.1.3 Plywood material

Plywood is a composite material, although we often consider it as a traditional working material. It is composed of individual plies / veneers of wood. It is very strong due to the way the plies are put together. The grain of each ply is positioned at ninety

degrees to the pieces of ply above and below it. The plies are glued together with synthetic resin, making a very strong composite material. Furthermore, plywood is usually constructed so that an odd number of plies are used. Plywood is less likely to warp or split, due to this construction. Manmade boards of this type are supplied in a range of sizes and thicknesses. This is an advantage compared to natural woods, as manmade boards can be manufactured so that they are extremely wide. This makes plywood a popular material in the construction industry. Softwood ply tends to be used in the construction industry for walls, roofs, and floors. Hardwood ply tends to be used quality laminate flooring, kitchen units, and some furniture. Marine plywood is used in boat hull construction. It is specially treated so that it is water resistant. Different samples of plywood are shown in Fig. 1.3.

Softwood plywood Hardwood plywood Marine plywood

Fig. 1.3: Different samples of plywood.

CHAPTER 2

BACKGROUND

2.1 Image Processing

In electrical engineering and computer science, image processing is any form of signal processing for which the input is an image, such as a photograph or video frame; the output of image processing may be either an image or, a set of characteristics or parameters related to the image.

2.1.1 Applications of image processing

Computer vision

Computer vision is the science and technology of machines that see, where see in this case means that the machine is able to extract information from an image that is necessary to solve some task [4].

Optical sorting

Optical sorting is a process of visually sorting a product though the use of photo-detectors (light sensors), cameras, or the human eye. In its simplest operation, a machine will simply see how much light is reflected off the object using a simple photo-detector (such as a photo-resistor) and accept or reject the item depending of how reflective it is (light or dark) [4].

Augmented reality

Augmented reality (AR) is a term for a live direct or indirect view of a physical real-world environment whose elements are augmented by virtual computer generated imagery. It is related to a more general concept called mediated reality in which a view of reality is modified (possibly even diminished rather than augmented) by a computer [4].

Face detection

Face detection is a computer technology that determines the locations and sizes of human faces in arbitrary (digital) images. It detects facial features and ignores anything else, such as buildings, trees and bodies. Feature detection. In computer vision and image processing the concept of feature detection refers to methods that aim at computing abstractions of image information and making local decisions at every image point whether there is an image feature of a given type at that point or not. The resulting features will be subsets of the image domain, often in the form of isolated points, continuous curves or connected regions [4].

Lane departure warning system

In road-transport terminology, a lane departure warning system is a mechanism designed to warn a driver when the vehicle begins to move out of its lane (unless a turn signal is on in that direction) on freeways and arterial roads. These systems are designed to minimize accidents by addressing the main causes of collisions: driving error, distraction, and drowsiness.

Non-photorealistic rendering

Non-photorealistic rendering (NPR) is an area of computer graphics that focuses on enabling a wide variety of expressive styles for digital art. In contrast to traditional computer graphics, which has focused on photorealism, NPR is inspired by artistic styles such as painting, drawing, technical illustration, and animated cartoons.

Microscope image processing

Microscope image processing is a broad term that covers the use of digital image processing techniques to process, analyze and present images obtained from a microscope. Such processing is now commonplace in a number of diverse fields such as medicine, biological research, cancer research, drug testing, metallurgy, etc.

Morphological image processing

Mathematical morphological (MM) is a theory and technique for the analysis and processing of geometrical structures, based on set theory, lattice theory, topology, and random functions. MM is most commonly applied to digital images, but it can be employed as well on graphs, surface meshes, solids, and many other spatial structures [4].

Remote sensing

Remote sensing is the small- or large-scale acquisition of information of an object or phenomenon, by the use of either recording or real time sensing device(s) that are wireless, or not in physical or intimate contact with the object (such as by way of aircraft, spacecraft, satellite, buoy, or ship).

Some applications of image processing are shown in Fig. 2.1.

Automatic face detection with OpenCV | Augmented Reality[4] | Roadway with lane markings

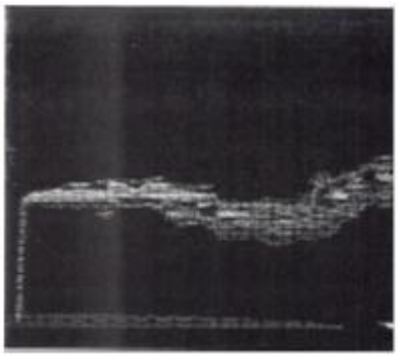
Fig. 2.1: Applications of image processing.

2.1.2 Image processing and civil engineering

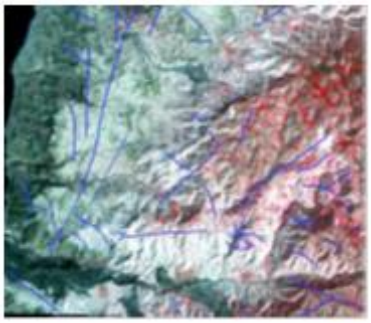
Over the three decades, in the civil engineering domain, image-processing methods have been applied in various areas. The main reasons to count on the advanced technology are due to such advantages as accuracy, objectivity, speed, and consistency. These distinct advantages have brought attention to state agencies to minimize the shortcomings of existing inspection practices.

Some applications of image processing for civil engineering are shown in Fig. 2.2.

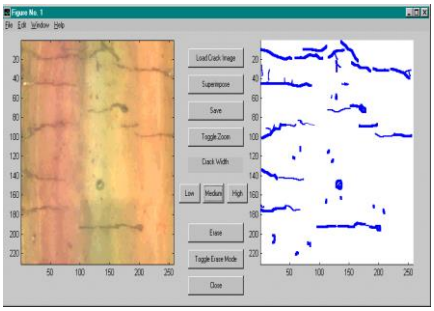
Bridge Coating Image with Defects (Image dimensions: 256x256) [27]



Application of image processing for wind engineering [45]



Application of image processing for delineate the structural features [13]



Application of image processing for crack detection [40]

Fig. 2.2: Applications of image processing for civil engineering.

A summary of various image processing applications that are used by civil engineers are presented in appendix A.

There has been growing interest in the applications of image processing in civil engineering. Several studies have been undertaken to study application of image processing in transportation fields. For example, image-processing techniques applied to road problems were investigated in Ref. [43]. In this paper, the areas chosen were road surface rating, measurement, and discrimination; calibrated sieve measurement; vehicle outline detection; and vehicle registration number recognition. This paper includes a review of the various image-processing methods available for invariant moment analysis and surface-texture discrimination and classification, and concludes that Laws masks are currently the preferred technique.

Ref. [31] used video image processing for traffic-data collection under mixed traffic conditions. For mixed traffic observed in developing countries, no suitable tool is available for analysis of traffic behavior. In this study, by keeping in view these necessities and problems in data collection, a novel offline image processing-based data collection system, suitable for mixed-traffic conditions, is developed.

(a different application?) The warning of rear-end accident on the highway based on image processing was studied by Ref. [29]. In this paper, the distance between the target car and the host car are calibrated with the help of binocular technology. Other parameters such as velocity and acceleration, were collected with other equipment, and are combined with the image-derived parameters. Subsequently, employing a clustering method, a classifying warning model that determines whether to accelerate or to decelerate is established so as to optimize driving safety on the highway. Using remote sensing, image processing and GIS techniques for transportation infrastructure and environmental capital asset management were investigated by Ref. [3]. The authors discuss important considerations for developing RS/GIS databases and analytic approaches, to enhance existing decision support systems, and present examples of the types of analysis that could be performed to integrate transportation infrastructure and environmental asset management. Ref. [19] used real-time image-processing algorithms for the detection of road and environmental conditions. This paper covered the design and implementation of an automated camera heading detection system to determine the directional components of a camera's position using the current camera image, various computer vision techniques, and a series of classification training images. Image-processing topics addressed as part of this research include edge detection, line detection, and two-dimensional filtering.

Also, some researchers have utilized image-processing techniques for geotechnical engineering fields. For example, Ref. [26] performed the analysis of rock fragmentation by using digital image processing. In this paper, a procedure for

calculating the size distribution of rock fragments using video images is described. The computer program first delineates the individual rock fragments in the images. This is followed by statistical procedures that takes into account fragment overlap and the two-dimensional nature of the images. Ref. [25] used numerical simulation and image processing for investigation of pore directivity of soils subjected to shearing. Ref. [21] performed accounting for void ratio variation in determination of grain size distribution by soil column image processing. The segregation is essential since grain size determination by image processing is presently only accurate for images that contain relatively uniform particles. For this purpose in this work, a soil sedimentation system was developed that allows sand grains to settle through water.

Some researchers utilized image-processing techniques for other civil-engineering fields. For example, Ref. [16] investigated a novel free-surface velocity measurement method by using spatio-temporal images. This paper describes first the underlying concept of the new measurement approach. Synthetic images are then processed using the proposed technique to verify the performance of the velocity measurement method. Ref. [30] studied the technical and computational aspects of the measurement of aggregate shape by digital image analysis. The system uses dual, synchronized, double speed progressive scan cameras to image the aggregate piece from two directions. In this paper, a dual-image acquisition card simultaneously digitizes both images and does real-time thresholding to create a binary image, which is ported to the host computer. Ref. [36] studied the reconnaissance of Golcuk 1999 earthquake damage using satellite images. This study utilizes remotely sensed pre- and post-disaster images in order to detect any change specifically associated with structural and major regional damage caused by natural disasters such as a strong earthquake. Ref. [46] studied the measurement method for continua by using image processing. In this paper, the measurement system for continua using the image processing technique is developed. At first, a correlation-based template that matches an Eulerian coordinate system is extended to trace the motion of

material points in a Lagragian coordinate system and its accuracy is updated by including deformation of the template in the matching procedure. Then, a correction method for the measured deformation field is proposed.

2.1.3 Image processing in construction engineering and management

A construction project is a complex development that typically occurs in an outdoor environment and involves a variety of workers, a range of equipment, and diverse materials. To successfully complete a given project, numerous business decisions must be made by all levels of management; sound decisions are based on a well-developed project scope and project plan, and, most crucially, the ongoing reassessment of the changing and dynamic condition of the construction field. However, the complex, shifting nature of construction makes it extremely challenging to assess quickly and thoroughly the status of the construction field at any particular time. For example, it is a daunting task to track the progress made across a range of physical components during the construction of a facility, let alone verify the quality of that construction on a regular basis.

Image processing has been extensively and successfully used in many sub-areas of civil engineering, such as engineering document scanning, pavement distress assessment, site evaluation via satellite imagery, studies of crack propagation and microstructure in cement-based materials, and evaluation of soil fabric, etc. It is a remarkably versatile tool that provides a means to augment existing methods of analysis and also opens up a large number of possibilities for significant advances in current civil engineering practices.

Although the geotechnical and transportation engineering sub-areas of civil engineering have taken the lead in applying image processing techniques to solve practical problems over the last ten years, the advancements in hardware and software

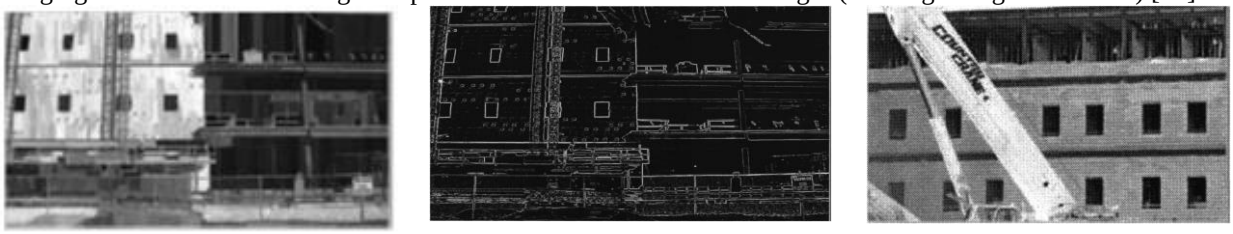
developed for digital image processing and analysis provide promising opportunities in construction management and site investigation applications.

Several studies have been undertaken to study application of image processing in construction engineering and management fields. For example, Ref. [47] performed a study on key techniques of image processing and automatic recognition of tunnel cracks. They found that on the surface of the tunnel research results is helpful to automatic identification of tunnel defects on the surface and reduction of cracks bearings on tunnel life. Ref. [42] studied an integrated digital image processing pavement management information system. This paper presented a method of integration of image processing and management features to the PMIS (Pavement Management Information System) software database. Imaging tools for evaluation of gusset plate connections in steel truss bridges was studied by Ref. [20]. Their results shown that technique that used in this paper provides a new tool for bridge engineers to quickly collect gusset plate geometry that can be used in connection evaluations and rating and can further enhance bridge-management tasks. Also, concrete column recognition in images and videos was studied by Ref. [49]. This paper presented a novel method of automated concrete column detection from visual data.

Material-based construction site image retrieval was studied by Ref. [6]. In their paper, the proposed solution comprises six steps. First, each image is decomposed into its basic features (color, texture, structure, etc.) by applying a series of filters through averaging, convolution, and other techniques. Second, the image is divided into regions using clustering, and third, the feature signatures of each cluster are computed. Fourth, the meaningful image clusters are isolated from the rest by comparing each cluster signature with the feature signatures of materials in a material knowledge base. Fifth, the identified materials are assigned to the original image as attributes, and sixth, those attributes are used to compute relevancies of images with other images in a construction image database. This research shown that the material-based construction site image

retrieval method can successfully answer material-based image queries by pre-identifying the materials in each image and comparing material signatures instead of image signatures. The method retains and enhances the advantage of user friendliness of the BRF approach while giving the engineer the opportunity to retrieve images in real time based on higher level, domain-specific concepts such as materials instead of the low-level concepts of color, texture, and structure. Moreover, this method addresses all of the issues and limitations of other methodologies previously discussed by taking advantage of the domain-specific characteristics of construction and overcoming the problem-specific deficiencies of the generic CBIR methods (e.g., low recall, focus on precision and wide-domain databases, etc.). Some applications of image processing for construction engineering and management are shown in Fig. 2.3.

Imaging tools for evaluation of gusset plates connections in steel truss bridges (tracking changes over time) [20]



(a) (b) (c)

a) Clustered image (each monochromatic area is a cluster), b) Sobel filter transformation (edges in white) and c) Image intensity [6]

Fig. 2.3: Applications of image processing for construction engineering and management.

2.2 Neural Network

Neural networks (NN) [etc….] appeal to many researchers due to their great closeness to the structure of the brain, a characteristic not shared by more traditional systems. In an analogy to the brain, an entity made up of interconnected neurons, neural networks are made up of interconnected processing elements called units, which respond in parallel to a set of input signals given to each. The unit is the equivalent of its brain counterpart, the neuron.

A neural network consists of four main parts:

1. Processing units $\{u_j\}$, where each u_j has a certain activation level aj(t) at any point in time.
2. Weighted interconnections between the various processing units which determine how the activation of one unit leads to input for another unit.
3. An activation rule which acts on the set of input signals at a unit to produce a new output signal, or activation.
4. Optionally, a learning rule that specifies how to adjust the weights for a given input/output pair.

A neural network is a powerful data-modeling tool that is able to capture and represent complex input/output relationships. Neural-network technology performs "intelligent" tasks similar to those performed by the human brain. It acquires knowledge through learning and then stores that knowledge within inter-neuron connection strengths known as synaptic weights. Neural networks can be applied to a wide variety of problems, from breast cancer detection to classification of satellite imagery. Below, we have included numerous examples of how NeuroSolutions and other NeuroDimension products can be used to apply neural network technology to real-world applications.

2.2.1 Types of neural network

2.2.1.1 Multilayer perceptron (MLP)

To be able to solve nonlinearly separable problems, a number of neurons are connected in layers to build a multilayer perceptron [5]. Each of the perceptrons is used to identify small linearly separable sections of the inputs.

Outputs of the perceptrons are combined into another perceptron to produce the final output. The hard-limiting (step) function used for producing the output prevents information on the real inputs flowing on to inner neurons. To solve this problem, the step function is replaced with a continuous function—usually the sigmoid function.

The Architecture of the multilayer perceptron

In a multilayer perceptron, the neurons are arranged into an input layer, an output layer and one or more hidden layers. The multilayer perceptron model is presented in Fig. 2.4.

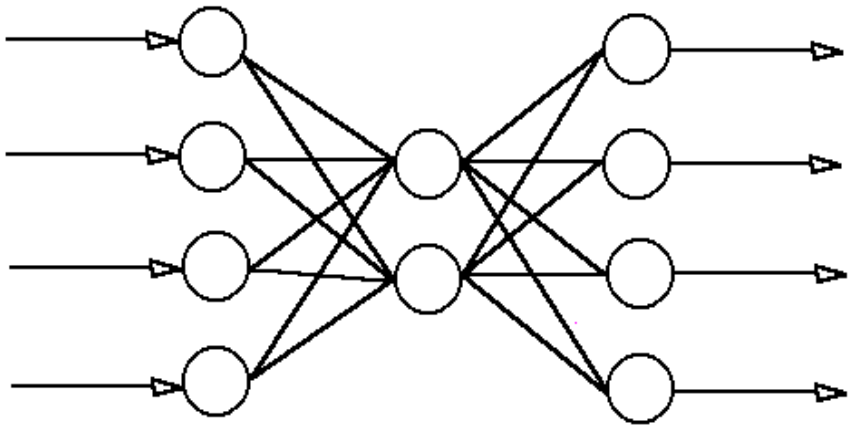


Fig. 2.4: The multilayer perceptron model.

Multilayer Perceptrons as Classifiers

The single layer perceptron is limited to calculating a single line of separation between classes. Let us consider a two-layer perceptron with two units in the input layer. If one unit is set to respond with a 1 if the input is above its decision line, and the other responds with a 1 if the input is below its decision line, the second layer produces a

solution in the form of a 1 if its input is above line 1 and below line 2. A 2-layer perceptron and the resulting decision region is presented in Fig. 2.5.

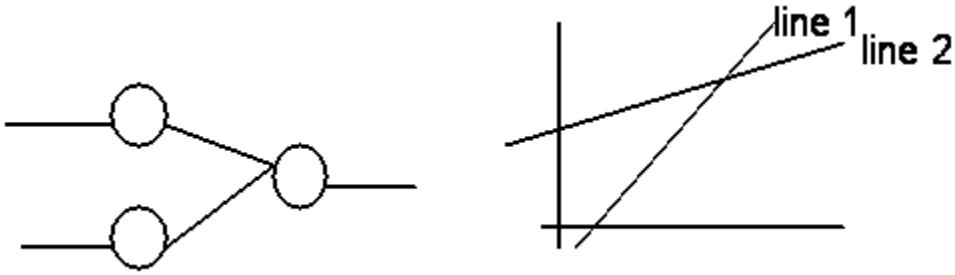


Fig. 2.5: A 2-layer perceptron and the resulting decision regio (something is missing in the caption.)

A three-layer perceptron can therefore produce arbitrarily shaped decision regions, and are capable of separating any classes. This statement is referred to as the Kolmogorov theorem. Considering pattern recognition as a mapping function from unknown inputs to known classes, any function, no matter how complex, can be represented by a multilayer perceptron of no more than three layers.

Advantages of Multilayer Perceptrons

The following two features characterise multilayer perceptrons and artificial neural networks in general. They are mainly responsible for the "edge" these networks have over conventional computing systems.

Generalisation

Neural networks are capable of generalisation, that is, they classify an unknown pattern with other known patterns that share the same distinguishing features. This means noisy or incomplete inputs will be classified because of their similarity with pure and complete inputs.

Fault Tolerance

Neural networks are highly fault tolerant. This characteristic is also known as "graceful degradation". Because of its distributed nature, a neural network keeps on

working even when a significant fraction of its neurons and interconnections fail. Also, relearning after damage can be relatively quick.

Applications of Multilayer Perceptrons

The multilayer perceptron with back propagation has been applied in numerous applications ranging from OCR (Optical Character Recognition) to medicine. Brief accounts of a few are given below.

Speech synthesis

A very well known use of the multilayer perceptron is NETtalk, a text-to-speech conversion system, developed by Sejnowski and Rosenberg in 1987.

It consists of 203 input units, 120 hidden units, and 26 output units with over 27000 synapses. Each output unit represents one basic unit of sound, known as a phoneme. Context is utilised in training by presenting seven successive letters to the input and the net learns to pronounce the middle letter. Ninety percent correct pronunciation achieved with the training set (80-87% with unseen set). It is also resistant to damage and displays graceful degradation. Multilayer perceptrons are also being used for speech recognition to be used in voice activated control systems.

Financial applications

Examples include bond rating, loan application evaluation, and stock market prediction. Bond rating involves categorising the bond issuer's capability. There are no hard-and-fast rules for determining these ratings. Statistical regression is inappropriate because the factors to be used are not well defined. Neural networks trained with back propagation have consistently outperformed standard statistical techniques.

Pattern Recognition

For many of the applications of neural networks, the underlying principle is that of pattern recognition. Target identification from sonar echoes has been developed. Given only a day of training (give the reference to what you are talking about), the net produced 100% correct identification of the target, compared to 93% scored by a Bayesian

classifier. There are many commercial applications of networks in character recognition. One such system performs signature verification on bank cheques.

Networks have been applied to the problems of aircraft identification, and to terrain matching for automatic navigation.

Computationally expensive learning process, no guaranteed solutions and scaling problems are the limitations of multilayer perceptrons.

2.2.1.2 Artificial neural network

An artificial neural network is a system based on the operation of biological neural networks, in other words, is an emulation of biological neural system. Why would be necessary the implementation of artificial neural networks? Although computing these days is truly advanced, there are certain tasks that a program made for a common microprocessor is unable to perform; even so a software implementation of a neural network can be made with their advantages and disadvantages. Some advantages and disadvantages of artificial neural network are summarized in appendix B.

Another aspect of the artificial neural networks is that there are different architectures, which consequently require different types of algorithms. Despite the superficial appearance of complexity, a neural network is relatively simple. In the world of engineering, neural networks have two main functions: pattern classifiers and as non-linear adaptive filters. As its biological predecessor, an artificial neural network is an adaptive system. By adaptive, it means that each parameter is changed during its operation and it is deployed for solving the problem in matter. This is called the training phase. An artificial neural network is developed with a systematic step-by-step procedure that optimizes a criterion commonly known as the learning rule. The input/output training data is fundamental for these networks as it conveys the information, which is necessary to discover the optimal operating point. In addition, a non-linear nature makes neural

network processing elements a very flexible system. The style of neural computation is presented in Fig. 2.6.

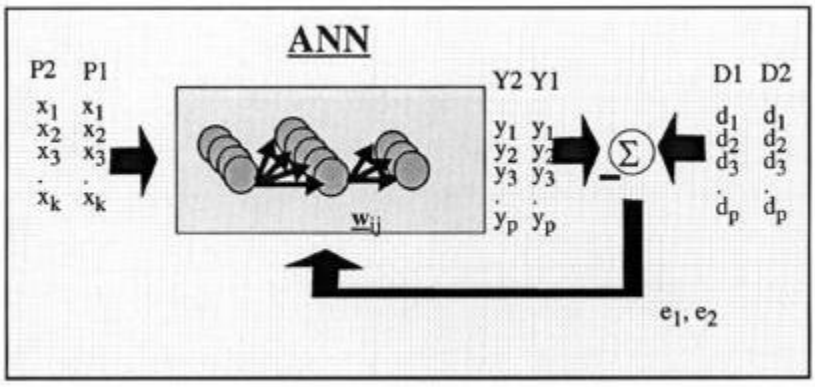


Fig. 2.6: The style of neural computation.

Basically, an artificial neural network is a system, *viz.* a structure that receives an input, process the data, and provides an output. Commonly, the input consists in a data array which can be anything such as data from an image file, a WAVE sound, or any kind of data that can be represented in an array. Once an input is presented to the neural network, and a corresponding desired or target response is set at the output, an error is composed from the difference of the desired response and the real system output.

The error information is fed back to the system that makes all adjustments to their parameters in a systematic fashion (commonly known as the learning rule). This process is repeated until the desired output is acceptable. It is important to note that the performance hinges heavily on the data. Hence, this is why this data should in practice be pre-process with third-party algorithms such as DSP algorithms.

In neural-network design, the engineer or designer chooses the network topology, the trigger function or performance function, learning rule, and the criteria for stopping the training phase. Thus, it may be quite difficult to determine the size and parameters of the network as there are no hard-and-fast prescriptive rules or formula to do so. In practice, trial and error provides rule-of-thumb guidance on the network design. The

problem with this method is when the system does not work properly it is difficult to refine the solution. Despite this, neural networks are often very efficient in terms of development, time, and resources. Through experience, we can develop the intuition whether artificial neural networks provide real solutions that are difficult to match with other technologies.

Fifteen years ago, Denker said, “artificial neural networks are the second best way to implement a solution.” This statement was motivated by their simplicity, design, and universality. Presently, neural-network technologies are emerging as the technology choice for many applications, such as patter recognition, prediction, system identification and control.

A radial basis function (RBF) network is an artificial neural network that uses radial basis functions as activation functions. The output of the network is a linear combination of radial basis functions of the inputs and neuron parameters. RBF networks always have three layers: input layer, hidden (radial basis) layer, and output layer (linear combination). Although the structure of RBF networks is simpler than MLP, RBF networks are often powerful in classification and regression problems.

2.2.1.3 Support vector machines

In machine learning, support vector machines (SVMs, also support vector networks) are supervised learning models with associated learning algorithms that analyze data and recognize patterns used for classification and regression analysis. The basic SVM takes a set of input data and predicts, for each given input, which of two possible classes forms the output, making it a non-probabilistic binary linear classifier. Given a set of training examples, each marked as belonging to one of two categories, an SVM training algorithm builds a model that assigns new examples into one category or the other. An SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide

as possible. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall on.

In addition to performing linear classification, SVMs can efficiently perform non-linear classification using what is called the kernel trick, implicitly mapping their inputs into high-dimensional feature spaces.

Advantages of support vector machines

SVM models have similar functional form to neural networks and radial basis functions, both popular data mining techniques. However, neither of these algorithms has the well-founded theoretical approach to regularization that forms the basis of SVM. The quality of generalization and ease of training of SVM is far beyond the capacities of these more traditional methods.SVM can model complex, real-world problems such as text and image classification, hand-writing recognition, and bioinformatics and biosequence analysis. SVM performs well on data sets that have many attributes, even if there are very few cases on which to train the model. There is no upper limit on the number of attributes; the only constraints are those imposed by hardware. Traditional neural nets do not perform well under these circumstances.

Support vector machine applications

In the last decade, support vector machines (SVMs) have increasingly turned into a standard methodology in the computer science and engineering communities [32]. Support vector machine has a various application in text categorization, bioinformatics, image recognition, handwritten digit recognition, image based gender identification, topic drift in page-ranking algorithms, etc.

2.2.2 Neural networks and civil engineering

Over the last few years, the use of neural networks has increased in many areas of civil engineering. These applications include geotechnical engineering, transportation engineering, structural analysis and design [8,17,38], structural damage assessment

[14,33], structural dynamics and control [15], and pavement condition-rating modeling [11]. Ref. [39] studied the artificial neural network (ANN) applications in geotechnical engineering. Artificial neural networks have been used successfully in pile capacity prediction, modeling soil behaviour, site characterisation, earth retaining structures, settlement of structures, slope stability, design of tunnels and underground openings, liquefaction, soil permeability and hydraulic conductivity, soil compaction, soil swelling and classification of soils. Ref. [39] provided a general view of some ANN applications for solving some types of geotechnical engineering problems. Also, this paper discussed the strengths and limitations of ANNs compared with the other modelling approaches. Application of neural network in civil engineering problems was studied by Ref. [24]. In this paper, an artificial neural network (ANN) is applied to several civil engineering problems, which have difficulty to solve or interrupt through conventional approaches of engineering mechanics. These include tide forecasting, earthquake-induced liquefaction and wave-induced seabed instability. Neural network assessment for scour depth around bridge piers was investigated by Ref. [23]. In this study, an alternative approach, artificial neural networks (ANN), is proposed to estimate the equilibrium and time dependent scour depth with numerous reliable database.

2.2.3 Neural networks and construction engineering and management

Applications of NN (Neural Networks) in construction management date to just the last few years (decade?). These applications cover a very wide area of construction issues. NN models have been developed internationally to assist the managers or contractors in many crucial construction decisions. Some of these models were designed for cost estimation, decision making, predicting the percentage of mark up, and predicting production rate [12]. Example for applications of NN in construction management are presented in Fig. 2.7.

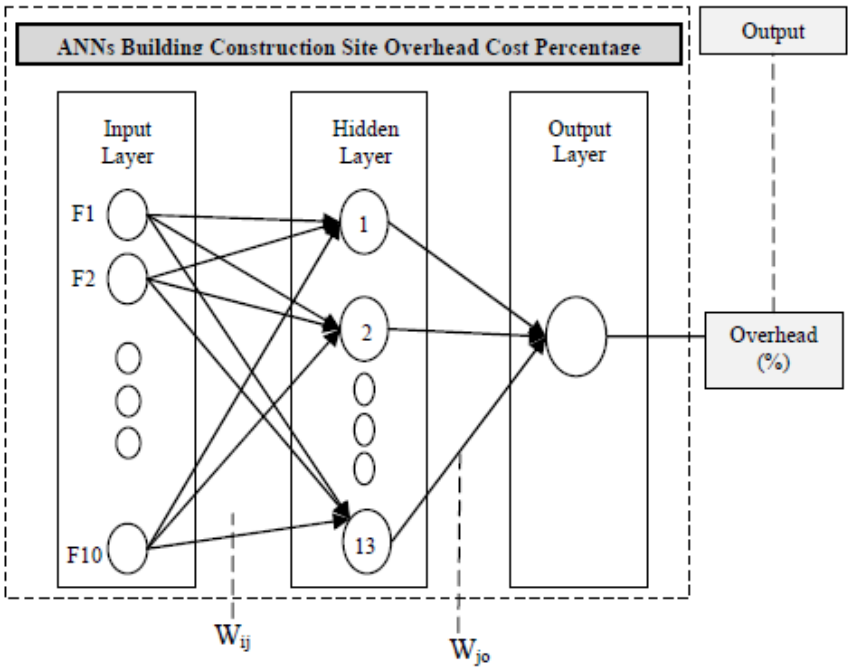


Fig. 2.7: Example for applications of neural networks in construction management [12].

The importance and influence of neural network NN etc. for construction engineering and management is supported by scores of recent studies. Neural network modeling of highway construction costs was studied by Ref. [44]. The objective of this research was to develop a procedure that estimates the escalation of highway construction costs over time. An artificial neural network model was developed which relates overall highway construction costs, described in terms of a highway construction cost index, to the cost of construction material, labor, and equipment, the characteristics of the contract and the contracting environment prevailing at the time the contract was let. Ref. [10] used a hybrid neural network for predicting construction labour productivity. This paper presents a novel computational intelligence (CI) approach to model construction labour productivity. A hybrid neural network combining the general regression neural network (GRNN), fuzzy logic (FL), and genetic algorithms (GA) are used to identify and quantify factors affecting construction labour productivity and to predict performance. Ref. [7]

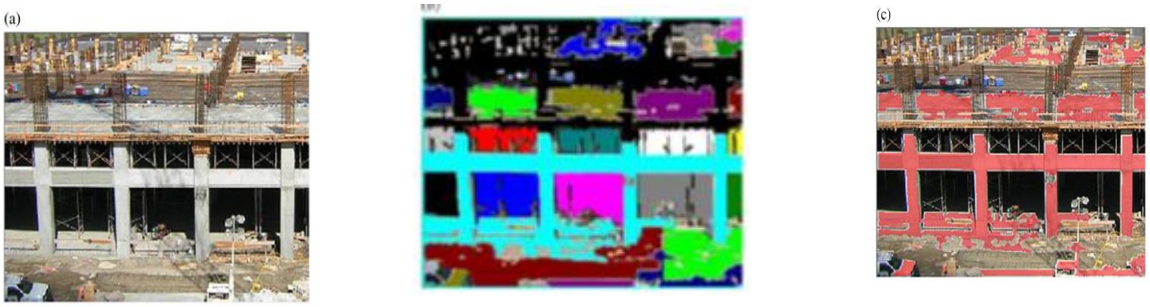
presented a NN and observation-data-based approach to estimating construction operation productivity. The main reason for using neural networks NN for construction productivity estimation is the requirement of performing complex mapping of environment and management factors to productivity. Test results show that the NN approach can produce a sufficiently accurate estimate with a limited data-collection effort, and thus has the potential to provide an efficient tool for construction productivity estimation. Ref. [2] used regularization neural network for construction cost estimation. In this paper, a regularization NN is formulated and a NN architecture is presented for estimation of the cost of construction projects. The model is applied to estimate the cost of reinforced-concrete pavements as an example. The new computational model is based on a solid mathematical foundation making the cost estimation consistently more reliable and predictable.

Ref. [35] used a neurofuzzy genetic system for selection of construction project managers. In this paper, the proposed fuzzy system was based on IF-THEN rules; a genetic algorithm improves the overall accuracy as well as the functions used by the fuzzy system to make initial estimates of the cluster centers for fuzzy c-means clustering. Moreover, a back-propagation neutral network method was used to train the system. The optimal measures of the inference parameters were identified by calculating the system's output error and propagating this error within the system. Also, Ref. [22] used an automated procedure for selecting project manager in construction firms. In previous work, the authors proposed a fuzzy system [35]. But in this paper, a simpler robust model is presented. The advantages of this new model lie in its simplicity and the fact that it is not necessary to consider many criteria in the selection procedure when using this model.

Ref. [41] investigated an automated color model-based concrete detection using machine learning algorithms in construction site images. This paper examined the feasibility of both generative and discriminative classifiers, Gaussian mixture models (GMMs), artificial neural networks (ANNs), and support vector machines (SVMs) in two

invariant color spaces to model the concrete color and classify concrete and non-concrete classes. For evaluation purposes, a database of 87 million pixels including concrete and non-concrete pixels was collected from 50 different construction sites of different concrete containing varying surface colors, with digital photographs taken under a variety of conditions. Then the three different classifiers in two invariant color spaces were compared to each other through the calculation of accuracy rates. The results from the comparative analysis indicate that SVM classifier in the HSI color space yielded the best detection performance, producing the highest accuracy rate. The verification results show that the proposed method is accurate and reliable for concrete detection, giving a consistent set of results.

Parameter optimization for automated concrete detection in image data was investigated by Ref. [48]. Their paper presented a novel automated method for detecting concrete regions in construction site images. Under the method, the image is first divided into several regions. The color/texture of each region is then extracted and input into a machine learning based classifier. The classifier is trained over one hundred samples and tested by another hundred samples. The output value indicates whether the region is composed of concrete. This way, concrete regions in a construction site image can be detected without the necessity of manually defining thresholds. The method presented in this paper was implemented. A prototype was developed using Microsoft Visual Studio C++ 2005. Hundreds of positive/negative concrete samples were used to train and test three types of classifiers (SVDD, C-SVC and ANN). The ANN based classifier is selected for concrete region identification. A database of real construction site images was used to test the validity of the overall method for concrete region detection. Some applications of neural network for construction engineering and management are shown in Fig. 2.8.



A concrete region detection example: (a) original image; (b) image segmentation; and (c) detection result [48]

Left: Construction site images, Right: Concrete detection results [41]

Fig. 2.8: Applications of neural network for construction engineering and management.

CHAPTER 3

RESEARCH METHODOLOGY

3.1 The Proposed Method

Detection in image processing means to check availability and determine the location of a particular object in an image. The concept of detection varies with recognition. The similarity of the detection and recognition is performing a similar classifier process. For detection, the goal is finding the location of a particular object among other objects and the background. Therefore, in the image-detection process by using a sliding window, the image is searched and at each step of the search, a binary classification is performed which determines whether the target object exists in the desired window. Therefore, in image detection, we seek to catagorise the image classifier into several specified classes. The detection process is usually more complicated than the recognition process. In the present research, we are sought to detect the masonry in the image. The proposed method for detecting masonry consists of two main steps: feature extraction and detection.

In feature extraction, after blocking the image, color, and texture features of the image in the format of a feature vector is extracted from each block. Next, the feature vectors are checked by a classifier to perform detection. Block diagram of the proposed method is shown in Fig. 3.1.

In the present research, we intend that perform a comparison between the performance of three classifiers for detection of concrete, brick, and plywood. The employed classifiers in this research are included as follows: multilayer perceptron (MLP), radial basis function (RBF), and support vector machine (SVM). For this purpose, feature vectors extracted from image blocks by all these methods can be

classified to perform a comparison between the efficiency of these methods for masonry detection.

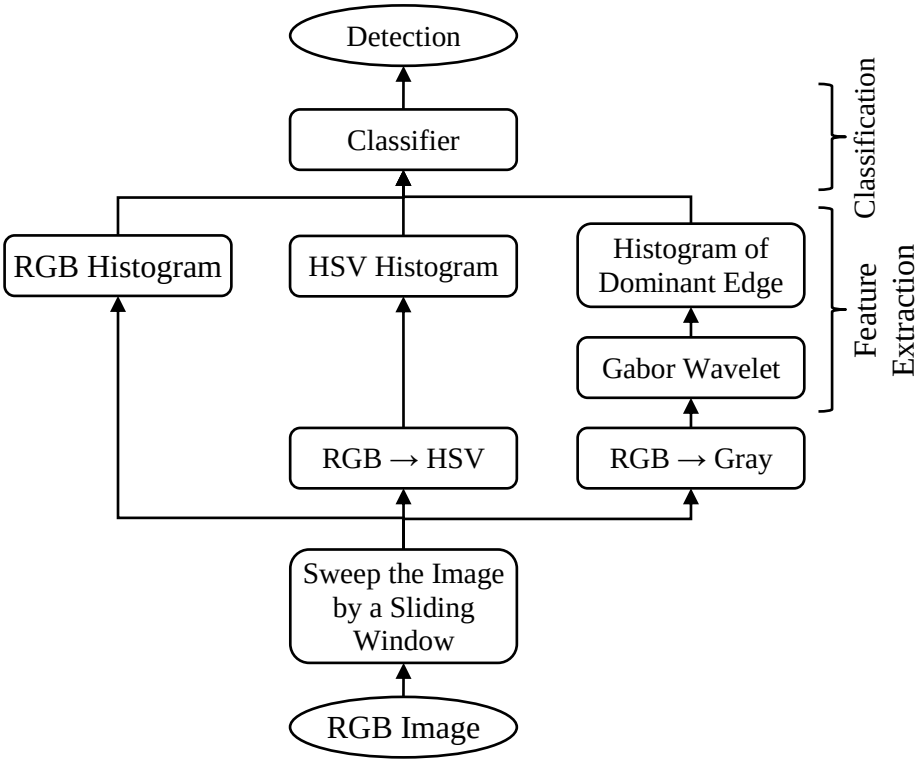


Fig. 3.1: Block diagram for proposed method.

Feature extraction

Within the proposed method for recognizing building material, instead of pixel-based processing approach, we follow a block-based processing approach. In pixel-based approaches, the extracted information is based on the information of each pixel separately; however, for block-based approaches, a block of pixels is considered and the decisions about whether or not the block is of specific type of material, are made by the information of all pixels of the block. The reason behind implementing a block-based approach for recognizing construction material is that construction material images mostly consist of uniform, connected pixels so it is more effective to consider a group of pixels instead of processing each pixel individually.

In the proposed block-based approach, each block of a picture consists of m×m pixels. Desired features are extracted from this block and are passed to a classifier for detection and recognition of construction material. As the next step, it is possible to generalize the results of recognizing the block to all of its pixels. In this case the image is divided into n×n overlapped blocks each one having m×m pixels. This approach is very fast yet inaccurate. The reason is that blocks may not always including the relevant construction-material texture and there might be other irrelevant pixels in the block; e.g., pixels belong to the background, sky, and so on.

The other approach would be using the obtained results from one block only for the central pixel of that block. In this case the image is divided into n×n overlapping blocks each one having m×m pixels. Due to the overlap between the blocks, this approach is computationally more expensive; however, the level of accuracy of the results is higher than the previous approach.

The suitable value of m depends on nature of the recognition problem as well as image dimensions. By increasing the value of m, more pixels are used for recognition purposes and there is the chance that some of the pixels in the block entail different information and thus, are not compatible with the rest of the pixels. On the other hand, decreasing the value of n is equal to utilizing fewer pixels for processing so the results might not be sufficiently accurate. As a result, the problem of choosing proper value for m is a trade-off problem and should be handled by a trial and error approach. In this research, the size of blocks is considered as 50×50 pixels.

In the proposed method, three distinct features of each block are extracted for further processing: RGB histogram, HSV histogram, histogram of dominant edges,

The first and second features describe the color properties of each block while the third feature represents the texture properties of blocks. The first feature is the histogram of the block’s pixels in RGB space. To compute the histogram, the histogram of each red,

green, and blue channels are independently represented in an 8 bit channels so the RGB histogram would be a 24 dimensional vector.

The second category of features is the HSV histogram of the block. To compute this histogram, as the first step, the color values are converted from RGB space into HSV space and following the same procedure explained for RGB histogram, the HSV histogram of each block is computed. Similar to RGB space, the HSV histogram of each block is a 24 dimensional vector.

The third group of features includes dominant edge histograms. To extract the dominant edge histogram, the image edges should be extracted in different orientations using different wavelets. To extract the edges, we use the 2D Gabor wavelet. The most significant advantage of 2D Gabor wavelet, compared to 2D discrete wavelet transforms (2D DWT) lies in its ability for extracting edges in different orientations and widths.

To extract the edges, the Gabor wavelet was implemented in four directions, $\left\{0, \frac{\pi}{4}, \frac{\pi}{2}, \frac{3\pi}{4}\right\}$ and three widths $\{2,4,6\}$. For this reason, 12 different Gabor wavelets were convolved with the pictures in different orientations and with different widths. As aresult, for each pixel of the image, 12 Gabor coefficients are calculated. Each of these coefficients represents the edge energy in that specific orientation. As the next step, we calculate the energy levels for all 12 Gabor coefficients and select the coefficient with the highest energy level. If the value of this Gabor coefficient is higher than a threshold (50), then the associated pixel includes a dominant edge whose orientation and width are represented by the Gabor coefficient.

Following the same procedure, all dominant edges of the entire block can be extracted and as the next step, the extracted dominant edges are considered as the third group of features. In some cases, the Gabor coefficients might be less than the threshold value, therefore that pixels does not include any specific dominant edge. As the result, considering the fact that there might be 12 different types of dominant edges, the

histogram of dominant edges include a 13 dimensional vector. The first 12 dimensions represent the dominant edges in different orientations with different widths while the 13th dimension, indicates the number of pixels that does not entail any dominant edges.

Based on what has been explained so far, for each block, 61 values are extracted as features. 48 of these values represent the color properties of the block while the rest of 13 values demonstrate the texture properties.

Classifier

As previously mentioned, in this thesis we will use three different methods include MLP, RBF, and SVM for classifier. These classifiers (MLP, RBF and SVM) are the most widely used in the field of pattern recognition. As mentioned in a brief definition of each classifier, these classifiers have a different structure and methodology. Therefore, comparison of performance and benefit of these classifiers in the masonry detection are interesting and useful. Descriptions of these classifiers were introduced in the previous chapter.

CHAPTER 4

EXPERIMENTAL RESULTS AND DISCUSSION

4.1. Experimental Results

The results are presented in three parts: Concrete, brick and plywood. In order to the evaluation of the performance of the methods, the values of the false positive rate (FPR) and false negative rate (FNR) for each detection method are calculated. For example, FPR is the error percentage for wrong detected pixels of concrete and FNR is percentage error for non-detected pixels of concrete. In other words, FPR shows the value of the method error that the parts of the non-concrete wrongly detected as the concrete. Also, FNR shows the percentage error for areas that are concrete but are not detected by the algorithms. MLP has very different structures for the number of hidden layers and the number of neuron in each hidden layer. Obviously the performance of each structure is different but evaluating the performance of all these structures is very time consuming. For this reason, for investigation of neural network different structures only a selection of MLP structures will be evaluated.

For neural-network training, even with the very same training data, the obtained MLP network may not be the same for different experiments; thus we perform the tests by using MLP repeated ten times independently. For each iteration, neural network is trained and tested using the same data set independently. The presented results of FPR and FNR for MLP are the average result of ten repetitions. RBF has only one parameter, i.e., that which specifies the radius of functions. In the tests, the effect of the radius function on the accuracy of RBF will be investigated.

SVM is a linear classifier by itself, but it can also be used for classification of the non-linear problems by using different kernel. In conducting the experiments, the effects of kernel type and kernel parameters on SVM performance will be investigated.

4.1.1 The results of concrete detection tests:

For performing the experiments, 600 training images were used including 100 concrete samples and 500 non-concrete samples. Some of the training samples are shown in Figs. 4.1 and 4.2.

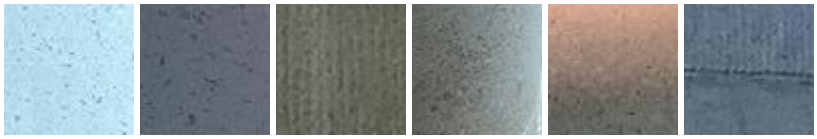


Fig. 4.1: Some of the concrete training samples.

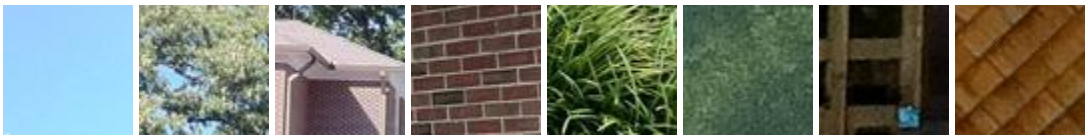


Fig. 4.2: Some of the non-concrete training samples.

Table 4.1: The results of concrete detection by using MLP and investigation of the MLP restructuring on performance.

Structure of Layers	FPR (%)	FNR (%)	Total Error (%)
[5]	5.24	46.82	52.06
[10]	2.59	44.14	46.73
[15]	3.49	44.09	47.58
[20]	3.14	43.74	46.88
[5,5]	4.21	50.89	55.1
[10,10]	4.64	46	50.64
[15,15]	3.16	47.61	50.77
[20,20]	1.64	50.28	51.92

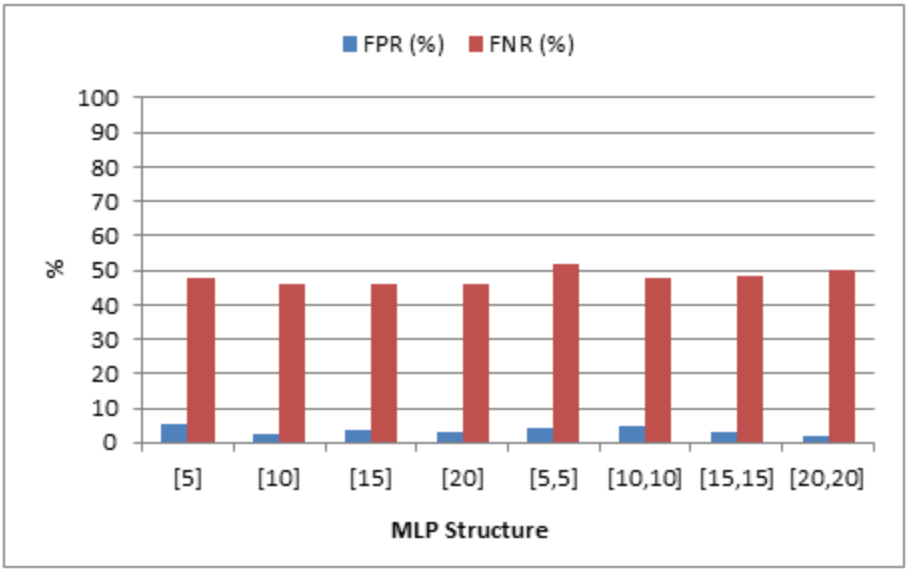


Fig. 4.3: Variation of FPR and FNR with MLP structure.

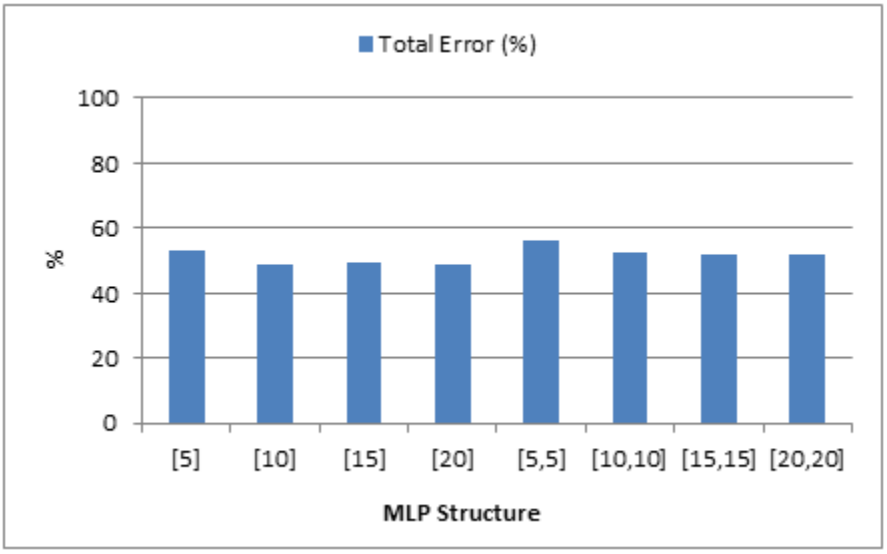


Fig. 4.4: Variation of total error with MLP structure.

Table 4.2: The results of concrete detection by using RBF and investigation of the radius functions changes on performance.

Radius	FPR (%)	FNR (%)	Total Error (%)
1	2.61	40.39	43
5	3.1	43.27	46.37
10	4.29	45.86	50.15
15	5.42	44.64	50.06
20	5.98	44.62	50.6
25	6.03	44.8	50.83

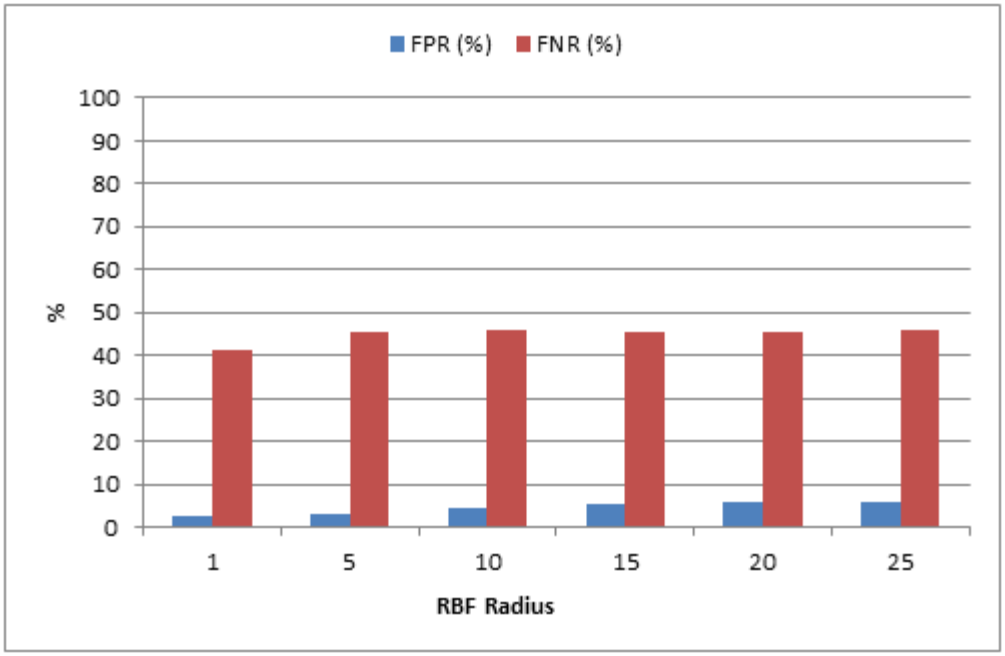


Fig. 4.5: Variation of FPR and FNR with RBF radius.

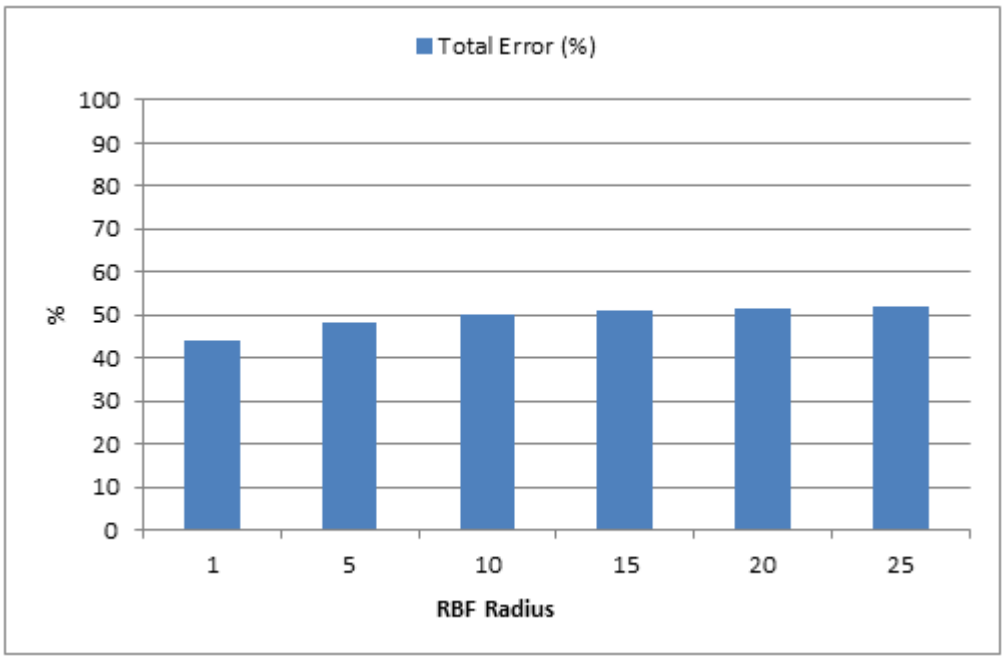


Fig. 4.6: Variation of total error with RBF radius

Table 4.3: The results of concrete detection by using SVM and investigation of the changing kernel type and kernel parameters on performance.

Kernel	Parameters	FPR (%)	FNR (%)	Total Error (%)
Polynomial	1	5.1	46.16	51.26
	2	2.47	43.56	46.03
	3	1.77	48.86	50.63
	4	1.38	51.68	53.06
	5	1.05	85.86	86.91
Quadratic	-	2.47	43.56	46.03
RBF	1	0	99.38	99.38
	5	4.32	36.83	41.15
	10	8.81	33.54	42.35
	15	13.18	32.78	45.96
	20	14.96	31.5	46.46
	25	18.62	32.32	50.94
MLP	[1,-5]	34.38	40.41	74.79
	[1,-4]	39.64	41.23	80.87
	[1,-3]	39.65	40.65	80.3
	[1,-2]	41.29	34.88	76.17
	[1,-1]	40.59	39.52	80.11
	[2,-1]	40.46	37.36	77.82
	[3,-1]	40.78	36.56	77.34
	[4,-1]	41.11	36.82	77.93
	[5,-1]	41.44	37.21	78.65

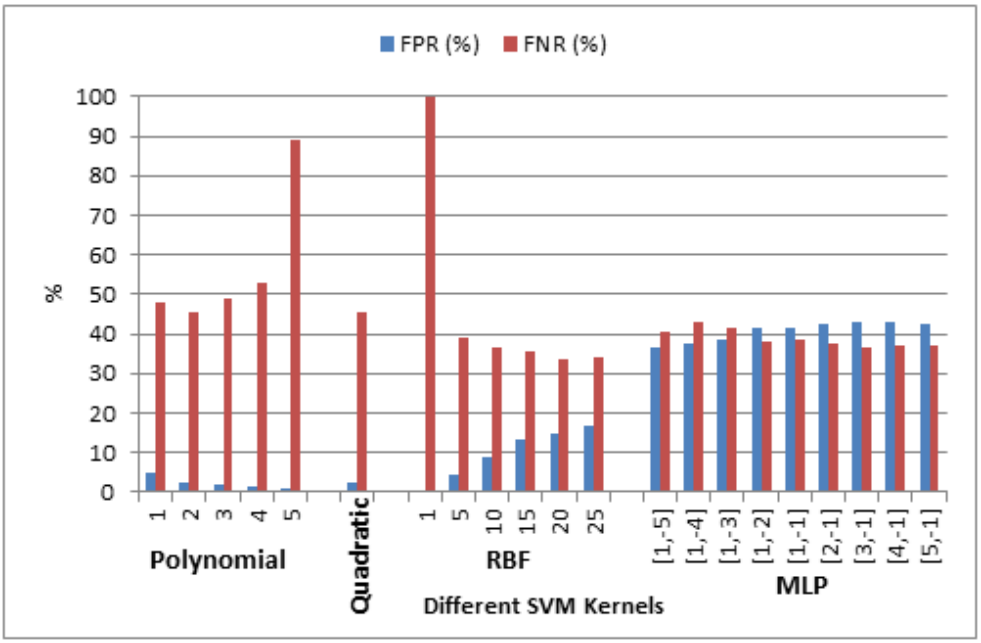


Fig. 4.7: Variation of FPR and FNR with kernel type and SVM kernel parameters.

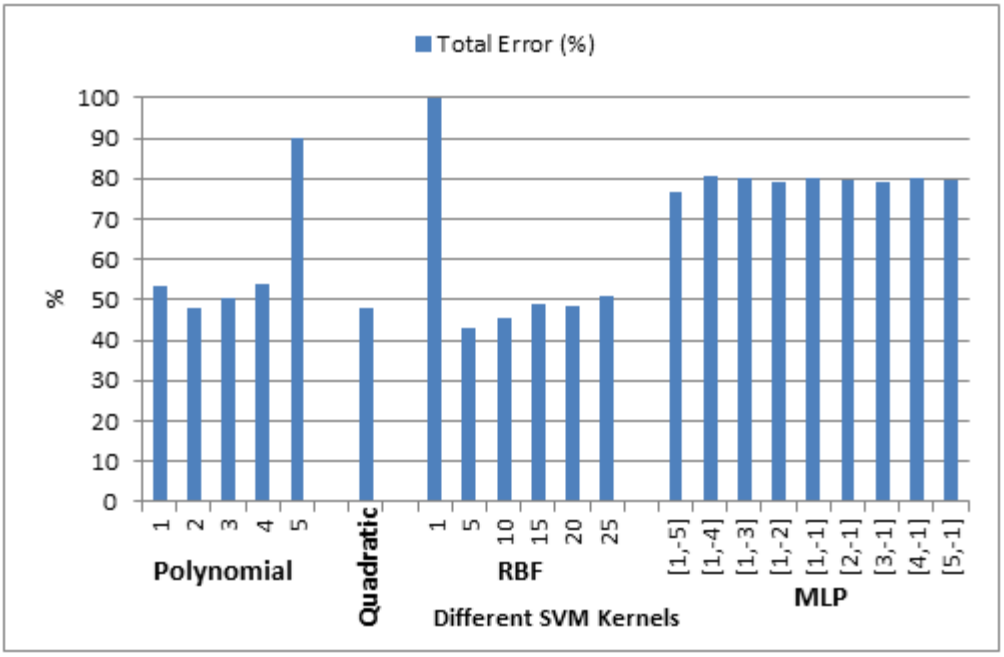


Fig. 4.8: Variation of total error with kernel type and SVM kernel parameters.

As can be seen, the error rates of the different classifiers (except SVM classifier with MLP kernel) are almost identical. Nevertheless, based on performed tests, the SVM classifier with the RBF kernel (with radius equal to 5) and the RBF classifier with radius equal to 1 have total error equal to 41.15% and 43%, respectively, and they gave the best performance.

Based on the results of the tests for concrete detection, in most cases the total error for concrete detection is in the vicinity of45% that it is due to the variability in the color and texture of the concrete. This is due to the camera distance from the target object and the surrounding colors are variable. As can be seen in Fig. 4.1, we expect that the concrete color to be gray but the effects of environmental color leads to the coloration of the concrete. Therefore, according to the objects color in the environment, the concrete color also can be changed. Also, if the concrete distance from the camera to be large, usually the texture of the concrete is uniform than the case that imaging is performed closer. Therefore, the texture of the concrete can be variable in different images. These changes in the stage of the training classifier and during use them for detection leads to

the error with a high percentage. The samples of brick detection by using different classifiers for some image are shown in the Fig. 4.9.

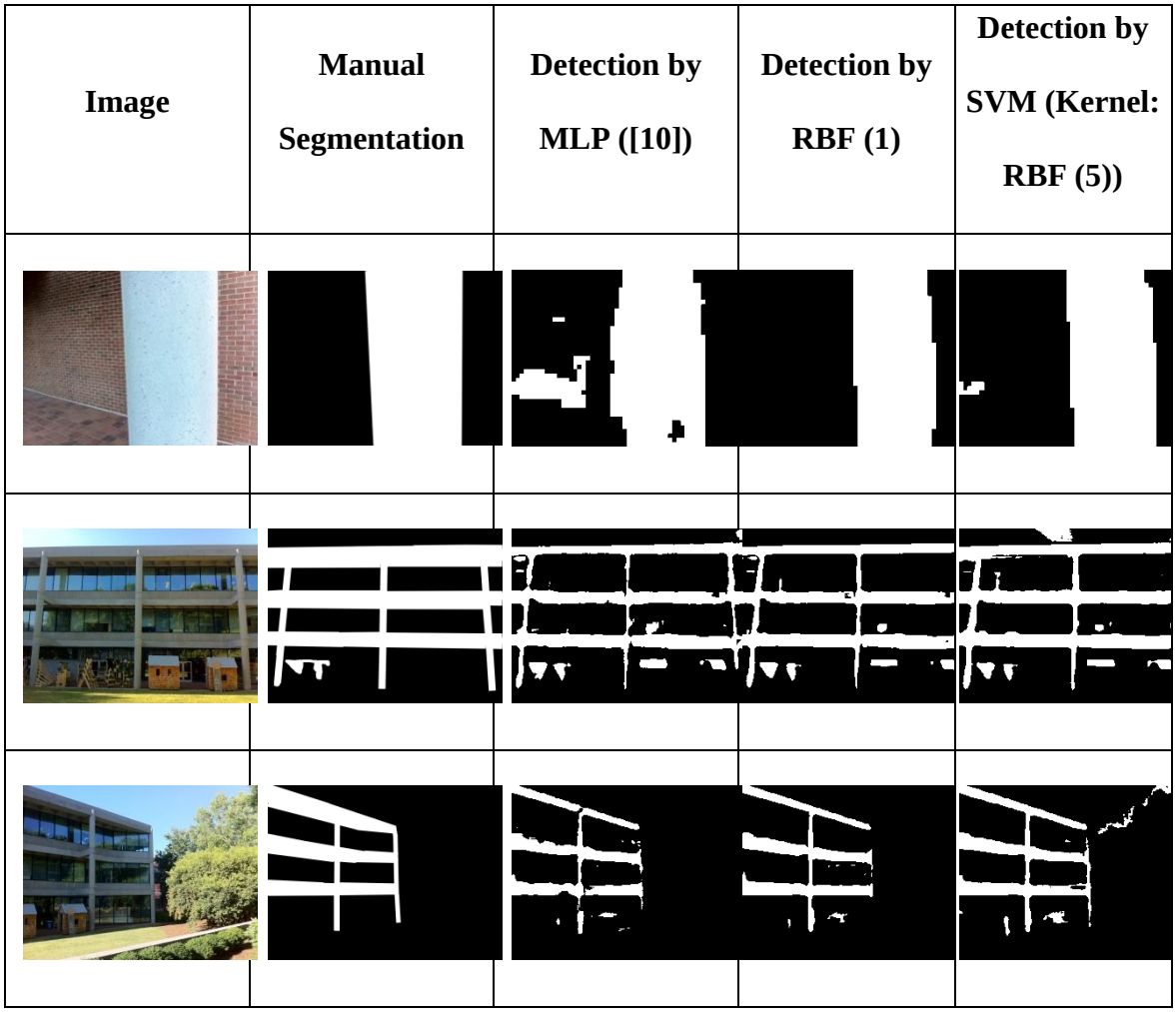


Fig. 4.9: The results of concrete detection by using SVM, RBF, and MLP for some image examples and comparison with manual segmentation.

4.1.2 The results of the brick detection tests

For performing the tests, 600 training images were used that included 100 brick samples and 500 non-brick samples. Some of the training samples are presented in Figs. 4.10 and 4.11.

Fig. 4.10: Some of the brick training samples.

Fig. 4.11: Some of the non-brick training samples.

Table 4.4: The results of brick detection by using MLP and investigation of the MLP restructuring on performance

Structure of Layers	FPR (%)	FNR (%)	Total Error (%)
[5]	8.77	13.86	22.63
[10]	4.65	14.8	19.45
[15]	5.16	18.91	24.07
[20]	6.14	16.44	22.58
[5,5]	5.15	21.78	26.93
[10,10]	3.75	18.43	22.18
[15,15]	3.62	22.27	25.89
[20,20]	3.95	21.87	25.82

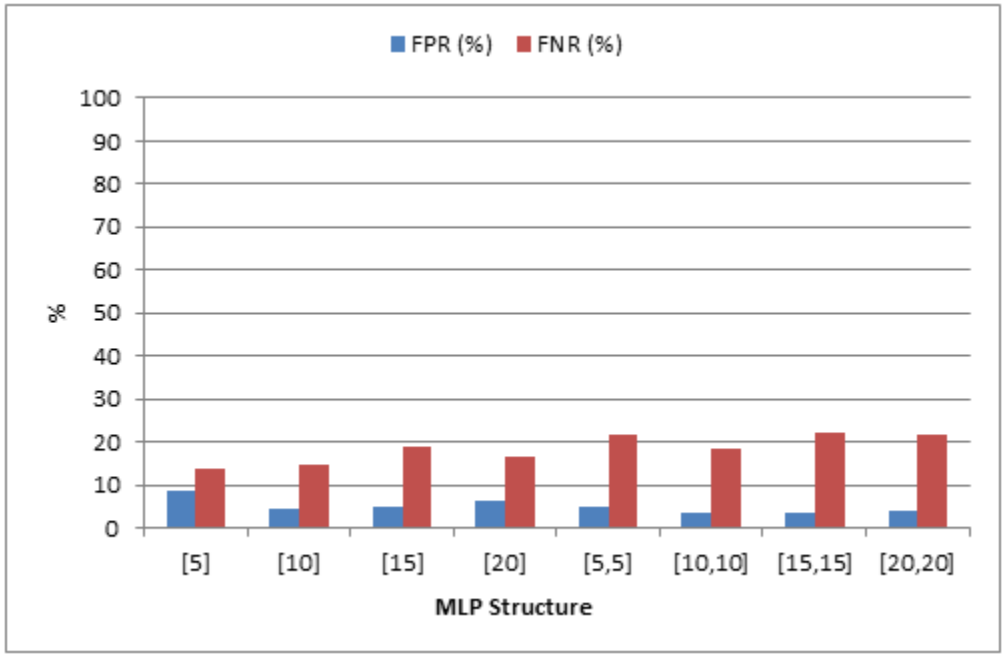


Fig. 4.12: Variation of FPR and FNR with MLP structure.

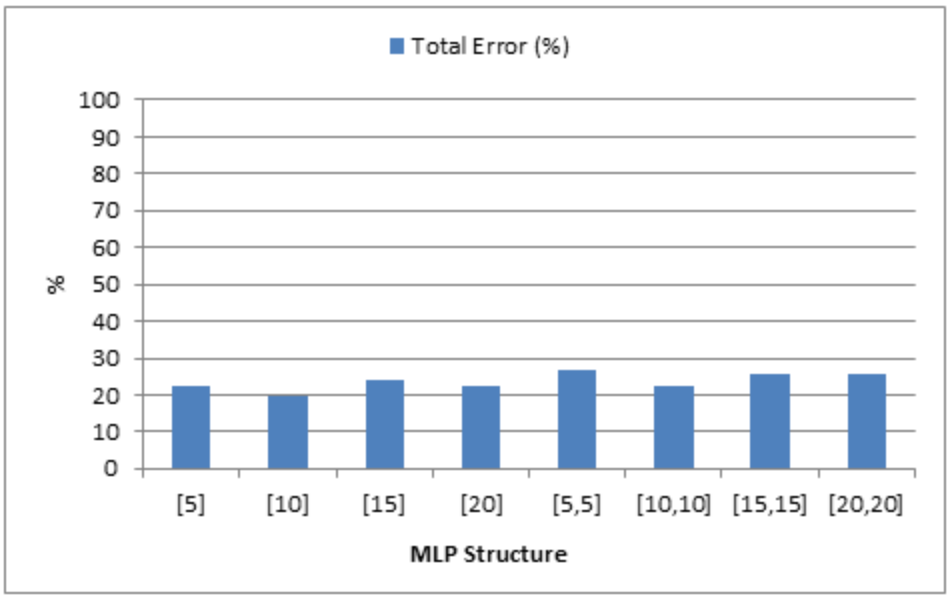


Fig. 4.13: Variation of total error with MLP structure.

Table 4.5: The results of brick detection by using RBF and investigation of the radius changes on performance.

Radius	FPR (%)	FNR (%)	Total Error (%)
1	2.63	18.85	21.48
5	3.27	15.96	19.23
10	3.98	17.29	21.27
15	4.39	18.06	22.45
20	4.59	18.16	22.75
25	4.73	18.21	22.94

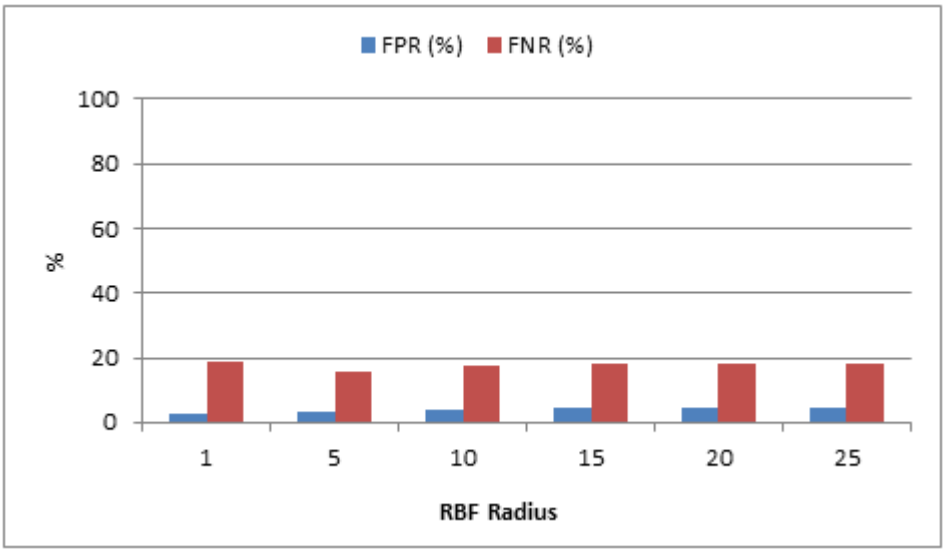


Fig. 4.14: Variation of FPR and FNR with RBF radius.

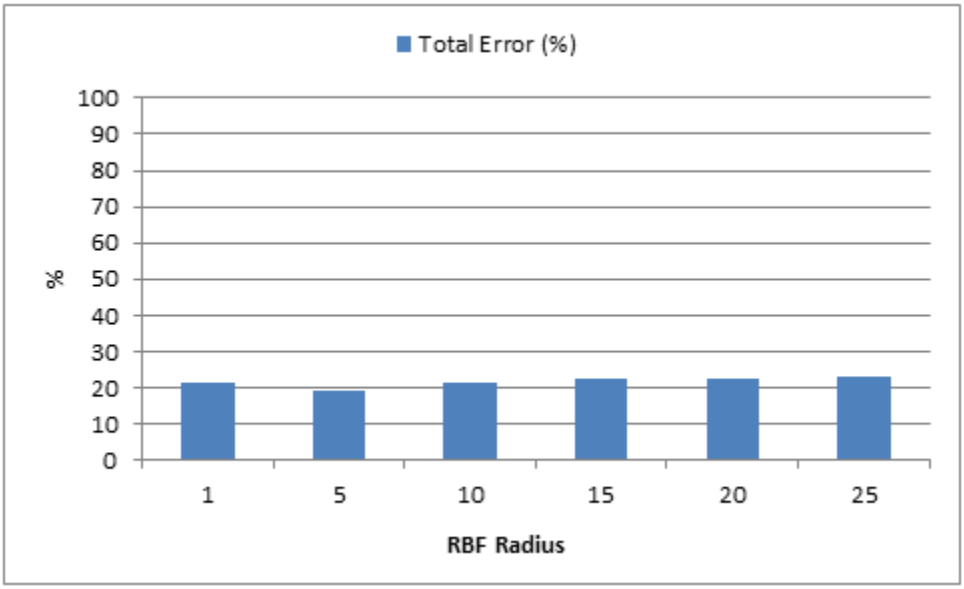


Fig. 4.15: Variation of total error with RBF radius.

Table 4.6: The results of brick detection by using SVM and investigation of the kernel type and kernel parameters changes on performance.

Kernel	Parameters	FPR (%)	FNR (%)	Total Error (%)
Polynomial	1	5.93	18.29	24.22
	2	3.92	15.3	19.22
	3	3.63	14.33	17.96
	4	5.07	17.19	22.26
	5	2.96	85.98	88.94
Quadratic	-	3.92	15.3	19.22
RBF	1	0	97.9	97.9
	5	3.57	16.07	19.64
	10	7.18	10.77	17.95
	15	7.96	9.09	17.05
	20	8.04	9.3	17.34
	25	8.17	9.23	17.4
MLP	[1,-5]	17.9	27.11	45.01
	[1,-4]	34.04	30.42	64.46
	[1,-3]	20.38	26.36	46.74
	[1,-2]	21.88	26.97	48.85
	[1,-1]	21.38	31.25	52.63
	[2,-1]	22.42	31.4	53.82
	[3,-1]	21.29	33.42	54.71
	[4,-1]	23.26	36.95	60.21
	[5,-1]	22.58	32.76	55.34

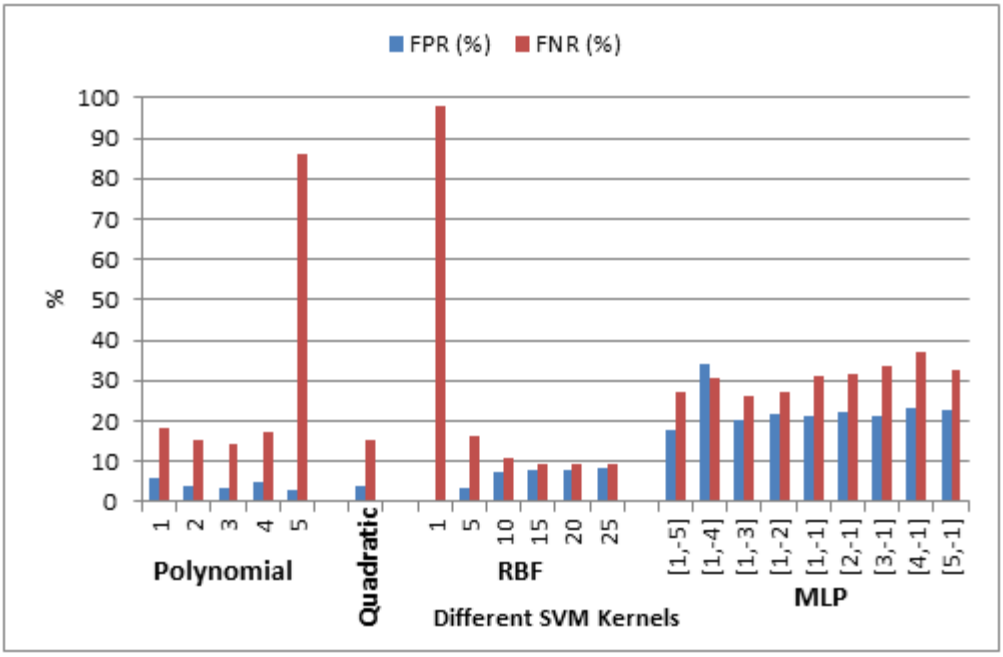


Fig. 4.16: Variation of FPR and FNR with kernel type and SVM kernel parameters.

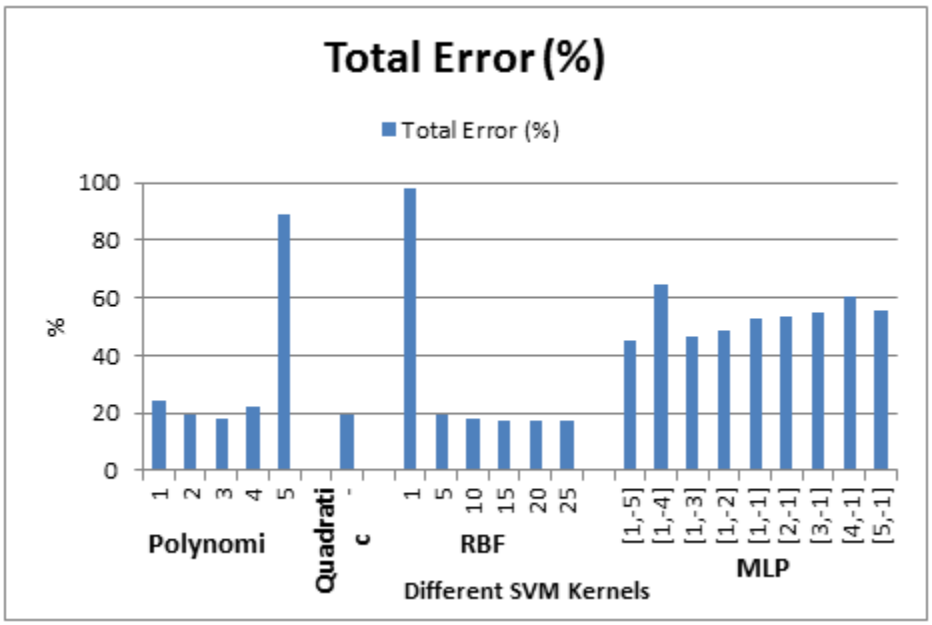


Fig. 4.17: Variation of total error with kernel type and SVM kernel parameters.

Based on the results of the tests, the error rates of the different classifiers (except SVM classifier with MLP kernel) are almost is identical and are in the vicinity of 20%. Based on the performed tests, SVM classifier with RBF kernel (with radiiof 10, 15, 20,

and 25) and polynomial kernel (third-order) with total error of~17% is the best classifier for the brick detection.

Based on the results of the tests for the brick detection, in most cases the total error for the brick detection is in the vicinity of 20% which is acceptable for several applications including visual progress monitoring and rough as-built documentation. The main reasons for increasing the error in brick detection are the lack of adequate lighting in the environment and blurring some part of the brick. In this case, brick color will change from red to black and it is difficult to detect. Also, by blurring some part of the brick, its texture will also indistinguishable. The samples of brick detection by using different classifier for some image are shown in the Fig. 4.18.

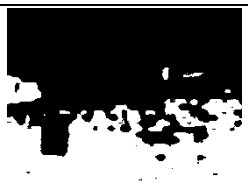
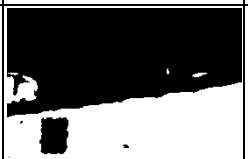
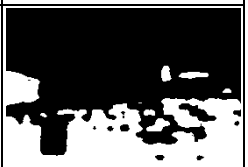
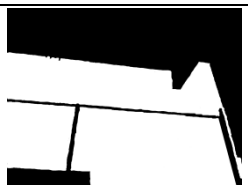
Image	**Manual Segmentation**	**Detection by MLP ([10])**	**Detection by RBF (5)**	**Detection by SVM (Kernel: RBF (15))**
				
				
				

Fig. 4.18: The results of brick detection by using SVM, RBF, and MLP for some image examples and comparison with manual segmentation.

4.1.3 The results of the plywood detection tests

For performing the tests, 600 training images were used that includes 100 plywood samples and 500 non-plywood samples.

Table 4.7: The results of plywood detection by using SVM and investigation of the changing kernel type and kernel parameters on performance.

Kernel	Parameters	FPR (%)	FNR (%)	Total Error (%)
Polynomial	1	5.64	24.19	29.83
	2	3.63	21.32	24.95
	3	3.11	21.23	24.34
	4	4.92	24.19	29.11
	5	2.72	92.53	95.25
Quadratic	-	3.62	20.31	23.93
RBF	1	0	98.1	98.1
	5	3.37	22.37	25.74
	10	6.92	16.37	23.29
	15	7.73	16.08	23.81
	20	7.94	16.14	24.08
	25	8.06	15.93	23.99
MLP	[1,-5]	17.34	32.14	49.48
	[1,-4]	33.93	36.52	70.45
	[1,-3]	20.14	36.26	56.4
	[1,-2]	21.68	32.37	54.05
	[1,-1]	21.17	37.45	58.62
	[2,-1]	22.21	37.84	60.05
	[3,-1]	21.22	39.52	60.74
	[4,-1]	23.14	43.85	66.99
	[5,-1]	22.24	38.25	60.49

Table 4.8: The results of plywood detection by using RBF and investigation of the radius functions changes on performance.

Radius	FPR (%)	FNR (%)	Total Error (%)
1	2.62	24.83	27.45
5	3.21	21.56	24.77
10	4.01	23.32	27.33
15	4.73	24.76	29.49
20	4.78	24.8	29.58
25	4.81	24.84	29.65

Table 4.9: The results of plywood detection by using MLP and investigation of the MLP restructuring on performance.

Structure of Layers	FPR (%)	FNR (%)	Total Error (%)
[5]	8.27	19.97	28.24
[10]	4.15	23.18	27.33
[15]	4.93	23.9	28.83
[20]	5.91	21.9	27.81
[5,5]	4.89	26.99	31.88
[10,10]	3.55	22.48	26.03
[15,15]	3.43	26	29.43
[20,20]	3.73	25.97	29.7

Based on the results of the tests, the error rates of the different classifiers (except SVM classifier with MLP kernel) are almost identical and are in the vicinity of 30%. Based on the performed tests, SVM classifier with RBF kernel (with radiiof 10, 15, 20, and 25) and polynomial kernel (third-order) with total error in the vicinity of 23% is the best classifier for the plywood detection.

Based on the results of the tests for the plywood detection, in most cases the total error for the plywood detection is in the vicinity of27% that it is acceptable for some applications. The main reasons for increasing the error in plywood detection are the lack of adequate lighting in the environment and blurring some part of the plywood.

CHAPTER 5

CONCLUSIONS AND FUTURE WORK

Digital images acquired at construction sites contain valuable information useful for different applications including As-built documentation of building elements, effective progress monitoring, structural damage assessment, and quality control of construction material. As the result there is an increasing need for effective methods for recognizing different building materials in digital images and videos. This paper presented a comparative analysis of three generative and discriminative machine learning algorithms including MLP, RBF, and SVM for recognizing construction materials in digital images. The results were presented for three class of building materials including concrete, plywood, and brick. The proposed method for detecting was including two steps: feature extraction and detection. In order to the evaluation of the performance of the methods, the value of the false positive rate (FPR) and the value of the false negative rate (FNR) for each detection method were achieved.

The results show that for concrete detection, in most cases the total error for concrete detection is in the vicinity of 50% that it is greatly because the color and texture of the concrete is variable. This is due to the camera distance from the target object and the surrounding colors are variable. For the brick detection, in most cases the total error for the brick detection is in the vicinity of 20% which is acceptable for some applications. The main reasons for increasing the error in brick detection are the lack of adequate lighting in the environment and blurring some part of the brick. Based on performed tests, the SVM classifier with RBF kernel (with radius of 5) and the RBF classifier with radius 1 are the best classifier for concrete detection. The main reason for this phenomenon is the complex texture of concrete in dataset. Also, the SVM classifier with RBF kernel (with radiuses equal to 10, 15, 20 and 25) and polynomial kernel (third-order) with total error in the vicinity of 17% is the best classifier for the brick detection. This is basically

due to more uniform texture of the brick. Based on the results of plywood detection, the SVM classifier with RBF kernel (with radiuses equal to 10, 15, 20 and 25) and polynomial kernel (third-order) with total error in the vicinity of 23% is the best classifier.

In summary obtained results reveal that by choosing right machine learning technique; it is possible to adequately detect different types of construction material in image data. Future work can perform on the other machine learning techniques (e.g. probabilistic neural network (PNN)) for recognizing construction materials. We can focus on the other building material such as plaster, steel, and stoneware. Also, the detection can perform by considering other information such as environmental conditions, and geometry of object.

APPENDIX A

APPLICATIONS OF IMAGE PROCESSING FOR CIVIL ENGINEERING

Application of image processing for wind engineering
Morphological Methods for Characterizing Inter-Feature Distances
Radiographic Visualization in Experimental Soil Mechanics
SURVEY: Fundamental Pavement Crack Imaging Algorithms
Machine Vision Based Traffic-Adjusted Intersection Signal Control
Image Analysis of Clay Microstructure
Displacement in the Soil Near a Tunnel Face using Centrifuge Tests and Image Processing
Digital Image Techniques for Volume Change Measurements in Triaxial Tests
Application of Digital Images Processing in Morphological Analysis of Geotextiles
Automated Crack Detection System Implementation in ARAN
Application of Radar Imaging Techniques to Concrete
Application on Digital Images and Processing for As-Built Construction Drawings

APPENDIX B

ADVANTAGES AND DISADVANTAGES OF NEURAL NETWORKS

Advantages:

It can be implemented without any problem
It can be implemented in any application
A neural network learns and does not need to be reprogrammed
When an element of the neural network fails, it can continue without any problem by their parallel nature
A neural network can perform tasks that a linear program cannot

Disadvantages:

The neural network needs training to operate
The architecture of a neural network is different from the architecture of microprocessors therefore needs to be emulated
Requires high processing time for large neural networks

APPENDIX C

SAMPLE CODE

Sample Code for Concrete Detection

- **Feature Extraction**

```
functionfeature_vector=FeatureExtraction(img,imgHSV,gabor_coef)
feature_vector=[];
% RGB Histogram
tmp=imhist(img(:,:,1),8);
feature_vector=[feature_vector;tmp];
tmp=imhist(img(:,:,2),8);
feature_vector=[feature_vector;tmp];
tmp=imhist(img(:,:,3),8);
feature_vector=[feature_vector;tmp];
% HSV Histogram
tmp=imhist(imgHSV(:,:,1),8);
feature_vector=[feature_vector;tmp];
tmp=imhist(imgHSV(:,:,2),8);
feature_vector=[feature_vector;tmp];
tmp=imhist(imgHSV(:,:,3),8);
feature_vector=[feature_vector;tmp];
% Wavelet based texture processing
tmp=DominantEdgeCountingWithThresholding(gabor_coef);
feature_vector=[feature_vector;tmp'];
% Feature Vector Normalization
imgSize(1)=size(img,1);
imgSize(2)=size(img,2);
feature_vector=feature_vector./(imgSize(1)*imgSize(2));
```

```
function histogram=DominantEdgeCountingWithThresholding(gabor_coef)
th=50;
gabor_coef_size=size(gabor_coef,3);
[mx,idx1]=max(abs(gabor_coef),[],3);
idx2=find(mx>=th);
histogram=hist(idx1(idx2),gabor_coef_size);
histogram=[(numel(idx1)-length(idx2)),histogram];
```

- **Gabor Wavelets**

```
functioncoef=ProcessingByGaborWavelet(img)
if length(size(img))==3
img=rgb2gray(img);
end
img=double(img);
lambda_array=[2,4,6];
theta_array=[0:pi/4:pi-0.001];
```

```
psi=0;
mask_size_coef=1;
coef_counter=0;
fori_theta=1:length(theta_array)
theta=theta_array(i_theta);
fori_lambda=1:length(lambda_array)
lambda=lambda_array(i_lambda);
sigma=2*lambda;
mask=GaborWavelet(lambda,theta,psi,sigma,mask_size_coef);
coef_counter=coef_counter+1;
coef(:,:,coef_counter)=imfilter(img,mask);
%coef{i_lambda,i_theta}=imfilter(img,mask);
%oriented_coef{i_theta}=oriented_coef{i_theta}+double(imfilter(img,mask
));
end
end
```

```
function mask=GaborWavelet(lambda,theta,psi,sigma,mask_size_coef)
x=[-mask_size_coef*sigma:mask_size_coef*sigma];
y=[-mask_size_coef*sigma:mask_size_coef*sigma];
[X,Y]=meshgrid(x,y);
XX=X*cos(theta)-Y*sin(theta);
YY=X*sin(theta)+Y*cos(theta);
mask=exp(-(XX.^2+YY.^2)/(2*sigma^2)).*cos(2*pi*XX/lambda+psi);
```

- **MLP main**

```
functionMLPMain

MLPTraining

path='Concrete-Image\';
file_list=dir([path,'*.jpg']);
fori=1:length(file_list)
tic
disp(['Image ',num2str(i),' out of ',num2str(length(file_list))]);
filename=file_list(i).name;
groundtruth_filename=strrep(filename,'.JPG','.bmp');
detected_filename=strrep(filename,'.JPG','_MLP.png');
imgRGB=imread([path,filename]);
img_groundtruth=imread([path,groundtruth_filename]);
img_groundtruth=boolean(img_groundtruth);
imgBW=MLPDetection(imgRGB);
imgBW=imresize(imgBW,size(img_groundtruth),'nearest');
    [fpr(i),fnr(i)]=DetectionEvaluation(img_groundtruth,imgBW);
disp(['FPR = ',num2str(fpr(i)),'    FNR = ',num2str(fnr(i))]);
imwrite(imgBW,[path,detected_filename]);
toc
end
fpr_avg=mean(fpr);
fnr_avg=mean(fnr);
```

```
disp(['Total FPR = ',num2str(fpr_avg),'    Total FNR =
',num2str(fnr_avg)]);

functionimgBW=MLPDetection(imgRGB)
ifnargin==0
imgRGB=imread('Concrete-Image\IMG_0839.jpg');
end
% Multi Layer Perceptron
imgSize(1)=size(imgRGB,1);
imgSize(2)=size(imgRGB,2);

loadMLPParameters.mat;

imgHSV=rgb2hsv(imgRGB);
gabor_coef=ProcessingByGaborWavelet(imgRGB);

patch_size=50;
step_size=10;

imgBW=zeros(imgSize);
row=1:step_size:imgSize(1)-patch_size;
col=1:step_size:imgSize(2)-patch_size;
fori=row
for j=col
sub_imgRGB=imgRGB(i:i+patch_size-1,j:j+patch_size-1,:);
sub_imgHSV=imgHSV(i:i+patch_size-1,j:j+patch_size-1,:);
sub_gabor_coef=gabor_coef(i:i+patch_size-1,j:j+patch_size-1,:);

feature_vector=FeatureExtraction(sub_imgRGB,sub_imgHSV,sub_gabor_coef);
output=sim(net,feature_vector);
imgBW(i,j)=(sign(output)+1)/2;
end
end
imgBW=imgBW(row,col);
imgBW=boolean(imgBW);
% imshow(imgBW);
```

- **MLP Detection**

```
functionimgBW=MLPDetection(imgRGB)
ifnargin==0
imgRGB=imread('Concrete-Image\IMG_0839.jpg');
end
% Multi Layer Perceptron
imgSize(1)=size(imgRGB,1);
imgSize(2)=size(imgRGB,2);

loadMLPParameters.mat;

imgHSV=rgb2hsv(imgRGB);
gabor_coef=ProcessingByGaborWavelet(imgRGB);

patch_size=50;
step_size=10;
```

```
imgBW=zeros(imgSize);
row=1:step_size:imgSize(1)-patch_size;
col=1:step_size:imgSize(2)-patch_size;
fori=row
for j=col
sub_imgRGB=imgRGB(i:i+patch_size-1,j:j+patch_size-1,:);
sub_imgHSV=imgHSV(i:i+patch_size-1,j:j+patch_size-1,:);
sub_gabor_coef=gabor_coef(i:i+patch_size-1,j:j+patch_size-1,:);

feature_vector=FeatureExtraction(sub_imgRGB,sub_imgHSV,sub_gabor_coef);
output=sim(net,feature_vector);
imgBW(i,j)=(sign(output)+1)/2;
end
end
imgBW=imgBW(row,col);
imgBW=boolean(imgBW);
% imshow(imgBW);
```

- **MLP Training**

```
functionMLPTraining
dataset_path_pos='concrete\';
dataset_path_neg='non-concrete\';
pos_training_img_no=65;
neg_training_img_no=210;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%
% Feature extraction
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%
pos_feature_vector=[];
disp('MLP Training ...');
fori=1:pos_training_img_no
%disp(['Positive Images: ',num2str(i),' out of
',num2str(pos_training_img_no)]);
img=imread([dataset_path_pos,num2str(i),'.jpg']);
imgHSV=rgb2hsv(img);
gabor_coef=ProcessingByGaborWavelet(img);
tmp=FeatureExtraction(img,imgHSV,gabor_coef);
pos_feature_vector=[pos_feature_vector,tmp];
end
neg_feature_vector=[];
fori=1:neg_training_img_no
%disp(['Positive Images: ',num2str(i),' out of
',num2str(neg_training_img_no)]);
img=imread([dataset_path_neg,num2str(i),'.jpg']);
imgHSV=rgb2hsv(img);
gabor_coef=ProcessingByGaborWavelet(img);
tmp=FeatureExtraction(img,imgHSV,gabor_coef);
neg_feature_vector=[neg_feature_vector,tmp];
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%
% Training
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%
feature_vector=[pos_feature_vector,neg_feature_vector];
class_label=[ones(1,size(pos_feature_vector,2)),-
ones(1,size(neg_feature_vector,2))];

net=feedforwardnet([20,20],'trainlm');
net.trainParam.epochs=100;
net.divideParam.trainRatio=0.8;
net.divideParam.valRatio=0.2;
net.divideParam.testRatio=0;
net=train(net,feature_vector,class_label);

output=sim(net,feature_vector);
output=sign(output);
error_rate=(sum(abs(output-class_label))/2)/length(output)

saveMLPParameters.matnet
```

- **RBF Main**

```
functionRBFMain
spread=20;

RBFTraining(spread)

path='Concrete-Image\';
file_list=dir([path,'*.jpg']);
fori=1:length(file_list)
tic
disp(['Image ',num2str(i),' out of ',num2str(length(file_list))]);
filename=file_list(i).name;
groundtruth_filename=strrep(filename,'.JPG','.bmp');
detected_filename=strrep(filename,'.JPG','_RBF.png');
imgRGB=imread([path,filename]);
img_groundtruth=imread([path,groundtruth_filename]);
img_groundtruth=boolean(img_groundtruth);
imgBW=RBFDetection(imgRGB);
imgBW=imresize(imgBW,size(img_groundtruth),'nearest');
    [fpr(i),fnr(i)]=DetectionEvaluation(img_groundtruth,imgBW);
disp(['FPR = ',num2str(fpr(i)),'    FNR = ',num2str(fnr(i))]);
imwrite(imgBW,[path,detected_filename]);
toc
end
fpr_avg=mean(fpr);
fnr_avg=mean(fnr);
disp(['Total FPR = ',num2str(fpr_avg),'    Total FNR =
',num2str(fnr_avg)]);
```

- **RBF Detection**

```
functionimgBW=RBFDetection(imgRGB)
ifnargin==0
imgRGB=imread('Image\IMG_0820.jpg');
end
```

```
% Multi Layer Perceptron
imgSize(1)=size(imgRGB,1);
imgSize(2)=size(imgRGB,2);

loadRBFParameters.mat;

imgHSV=rgb2hsv(imgRGB);
gabor_coef=ProcessingByGaborWavelet(imgRGB);

patch_size=50;
step_size=10;

imgBW=zeros(imgSize);
row=1:step_size:imgSize(1)-patch_size;
col=1:step_size:imgSize(2)-patch_size;
fori=row
for j=col
sub_imgRGB=imgRGB(i:i+patch_size-1,j:j+patch_size-1,:);
sub_imgHSV=imgHSV(i:i+patch_size-1,j:j+patch_size-1,:);
sub_gabor_coef=gabor_coef(i:i+patch_size-1,j:j+patch_size-1,:);

feature_vector=FeatureExtraction(sub_imgRGB,sub_imgHSV,sub_gabor_coef);
output=sim(net,feature_vector);
imgBW(i,j)=(sign(output)+1)/2;
end
end
imgBW=imgBW(row,col);
imgBW=boolean(imgBW);
% imshow(imgBW);
```

- **RBF Training**

```
functionRBFTraining(spread)
ifnargin==0
spread=1;
end
dataset_path_pos='concrete\';
dataset_path_neg='non-concrete\';
pos_training_img_no=65;
neg_training_img_no=210;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%
% Feature extraction
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%
pos_feature_vector=[];
disp('RBF Training ...');
fori=1:pos_training_img_no
%disp(['Positive Images: ',num2str(i),' out of
',num2str(pos_training_img_no)]);
img=imread([dataset_path_pos,num2str(i),'.jpg']);
imgHSV=rgb2hsv(img);
gabor_coef=ProcessingByGaborWavelet(img);
tmp=FeatureExtraction(img,imgHSV,gabor_coef);
pos_feature_vector=[pos_feature_vector,tmp];
```

```
end
neg_feature_vector=[];
fori=1:neg_training_img_no
%disp(['Positive Images: ',num2str(i),' out of
',num2str(neg_training_img_no)]);
img=imread([dataset_path_neg,num2str(i),'.jpg']);
imgHSV=rgb2hsv(img);
gabor_coef=ProcessingByGaborWavelet(img);
tmp=FeatureExtraction(img,imgHSV,gabor_coef);
neg_feature_vector=[neg_feature_vector,tmp];
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%
% Training
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%
feature_vector=[pos_feature_vector,neg_feature_vector];
class_label=[ones(1,size(pos_feature_vector,2)),-
ones(1,size(neg_feature_vector,2))];

% net=newrb(feature_vector,class_label,0,spread);
net=newrbe(feature_vector,class_label,spread);

output=sim(net,feature_vector);
output=sign(output);
error_rate=(sum(abs(output-class_label))/2)/length(output)

saveRBFParameters.matnet
```

- **SVM Main**

```
functionSVMMain
% kernel_function='linear';
% kernel_function='quadratic';
% kernel_function='polynomial';
% kernel_function='rbf';
kernel_function='mlp';

SVMTraining(kernel_function);

path='Concrete-Image\';
file_list=dir([path,'*.jpg']);
disp(['Kernel Function: ',kernel_function]);
fori=1:length(file_list)
tic
disp(['Image ',num2str(i),' out of ',num2str(length(file_list))]);
filename=file_list(i).name;
groundtruth_filename=strrep(filename,'.JPG','.bmp');

detected_filename=strrep(filename,'.JPG',['_SVM_',kernel_function,'.png
']);
imgRGB=imread([path,filename]);
img_groundtruth=imread([path,groundtruth_filename]);
img_groundtruth=boolean(img_groundtruth);
imgBW=SVMDetection(imgRGB);
```

```
imgBW=imresize(imgBW,size(img_groundtruth),'nearest');
    [fpr(i),fnr(i)]=DetectionEvaluation(img_groundtruth,imgBW);
disp(['FPR = ',num2str(fpr(i)),'    FNR = ',num2str(fnr(i))]);
imwrite(imgBW,[path,detected_filename]);
toc
end
fpr_avg=mean(fpr);
fnr_avg=mean(fnr);
disp(['Total FPR = ',num2str(fpr_avg),'    Total FNR =
',num2str(fnr_avg)]);
```

- **SVM Detection**

```
functionimgBW=SVMDetection(imgRGB)
ifnargin==0
imgRGB=imread('Image\IMG_0820.jpg');
end
% Multi Layer Perceptron
imgSize(1)=size(imgRGB,1);
imgSize(2)=size(imgRGB,2);

loadSVMParameters.mat;

imgHSV=rgb2hsv(imgRGB);
gabor_coef=ProcessingByGaborWavelet(imgRGB);

patch_size=50;
step_size=10;

imgBW=zeros(imgSize);
row=1:step_size:imgSize(1)-patch_size;
col=1:step_size:imgSize(2)-patch_size;
fori=row
for j=col
sub_imgRGB=imgRGB(i:i+patch_size-1,j:j+patch_size-1,:);
sub_imgHSV=imgHSV(i:i+patch_size-1,j:j+patch_size-1,:);
sub_gabor_coef=gabor_coef(i:i+patch_size-1,j:j+patch_size-1,:);

feature_vector=FeatureExtraction(sub_imgRGB,sub_imgHSV,sub_gabor_coef);
output=svmclassify(svm_strct,feature_vector');
imgBW(i,j)=(sign(output)+1)/2;
end
end
imgBW=imgBW(row,col);
imgBW=boolean(imgBW);
% imshow(imgBW);
```

- **SVM Training**

```
functionSVMTraining(kernel_function)
ifnargin==0
kernel_function='linear';
% kernel_function='quadratic';
% kernel_function='polynomial';
% kernel_function='rbf';
```

```
% kernel_function='mlp';
end
dataset_path_pos='concrete\';
dataset_path_neg='non-concrete\';
pos_training_img_no=65;
neg_training_img_no=210;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%
% Feature extraction
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%
pos_feature_vector=[];
disp('SVM Training ...');
fori=1:pos_training_img_no
%disp(['Positive Images: ',num2str(i),' out of
',num2str(pos_training_img_no)]);
img=imread([dataset_path_pos,num2str(i),'.jpg']);
imgHSV=rgb2hsv(img);
gabor_coef=ProcessingByGaborWavelet(img);
tmp=FeatureExtraction(img,imgHSV,gabor_coef);
pos_feature_vector=[pos_feature_vector,tmp];
end
neg_feature_vector=[];
fori=1:neg_training_img_no
%disp(['Positive Images: ',num2str(i),' out of
',num2str(neg_training_img_no)]);
img=imread([dataset_path_neg,num2str(i),'.jpg']);
imgHSV=rgb2hsv(img);
gabor_coef=ProcessingByGaborWavelet(img);
tmp=FeatureExtraction(img,imgHSV,gabor_coef);
neg_feature_vector=[neg_feature_vector,tmp];
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%
% Training
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%
feature_vector=[pos_feature_vector,neg_feature_vector]';
class_label=[ones(size(pos_feature_vector,2),1);-
ones(size(neg_feature_vector,2),1)];

switchkernel_function
case'polynomial'

svm_strct=svmtrain(feature_vector,class_label,'kernel_function','polyno
mial','polyorder',4);
case'rbf'

svm_strct=svmtrain(feature_vector,class_label,'kernel_function','rbf','
rbf_sigma',30);
case'mlp'

svm_strct=svmtrain(feature_vector,class_label,'kernel_function','mlp','
mlp_params',[1 -5]);
otherwise
```

```
svm_strct=svmtrain(feature_vector,class_label,'kernel_function',kernel_
function);
end

output=svmclassify(svm_strct,feature_vector);
output=sign(output);
error_rate=(sum(abs(output-class_label))/2)/length(output)

saveSVMParameters.matsvm_strctkernel_function
```

REFERENCES

[1] ACPA (American concrete pavement association, "Concrete material basics," www.pavement.com, 2013.

[2] ADELI, H., and WU, M. "Regularization Neural Network for Construction Cost Estimation," *Journal of Construction Engineering and Management*, vol. 124, pp. 18-24, 1998.

[3] AMEKUDZI, A., and BAFFOUR, R. "Using remote sensing, image processing and GIS techniques for transportation infrastructure and environmental capital asset management," *Presented at the Seventh International Conference on Applications of Advanced Technologies in Transportation (AATT)*, Boston Marriot, Cambridge, Massachusetts, USA, August 5-7, pp. 362-369, 2002.

[4] APTE, G., ZAREEN, S.K., and AGRAWAL, S.R. "Image processing-applications, typical operations and future trends," *BIOINFO Pattern Recognition*, vol. 1, pp. 5-7, 2011.

[5] BEALE, R., and JACKSON, T. "Neural computing: an Introduction," Adam Hilger, IOP Publishing Ltd: Bristol, 1990.

[6] BRILAKIS, I., SOIBELMAN, L., and SHINAGAWA, Y. "Material-based construction site image retrieval," *Journal of Computing in Civil Engineering*, vol. 19, pp. 341-355, 2005.

[7] CHAO, L-C., and SKIBNIEWSKI, M. "Estimating Construction Productivity: Neural-Network-Based Approach," Special Issue on Neural Networks, *Journal of Computing in Civil Engineering*, vol. 8, pp. 234-251, 1994.

[8] CLADERA, A., and MAR, A. "Shear design procedure for reinforced normal and high-strength concrete beams using artificial neural networks. Part I: beams without stirrups," *Journal Engineering Structure*, vol. 26, pp. 917-926, 2004.

[9] CLADERA, A., and MAR, A. "Shear design procedure for reinforced normal and high-strength concrete beams using artificial neural networks. Part II: beams with stirrups," *Journal Engineering Structure*, vol. 26, pp. 927-936, 2004.

[10] DISSANAYAKE, M., ROBIMSON FAYEK, A., RUSSELL, A., and PEDRYCZ, W. "A Hybrid Neural Network for Predicting Construction Labour Productivity," *Presented at the ASCE International Conference on Computing in Civil Engineering*, 12-15 July 2005, Cancun, Mexico, pp. 1-12, 2005.

[11] ELDIN, N., and SENOUCI, A. "A pavement condition-rating model using back propagation neural networks," *Microcomputers in Civil Engineering*, vol. 10, pp. 433-441, 1995.

[12] ElSAWY, I., HOSNY, H., and ABDEL RAZEK, M. "A neural network model for construction projects site overhead cost estimating in Egypt," *International Journal of Computer Science*, vol. 8, pp. 273-283, 2011.

[13] EYAD, H.R., KAKISH, M., and AL-HANBALI, N.N. "Utilizing remote sensing and digital image processing to delineate the structural features in the eastern Part of the dead Sea, Jordan," *Jordan Journal of Civil Engineering*, vol. 1, pp. 287-294, 2007.

[14] FENG, M., and BAHNG, E. "Damage Assessment of Jacketed RC Columns Using Vibration Tests," *Journal of Structure Engineering*, vol. 125, pp. 265-271, 1999.

[15] FENG, M., and KIM, J. "Identification of a Dynamic System Using Ambient Vibration Measurements," *Journal of Applied Mechanic*, vol. 65, pp. 1010- 1023, 1998.

[16] FUJITA, I., and TSUBAKI, R. "A novel free-surface velocity measurement method using spatio-temporal images," *Presented at the Hydraulic Measurements and Experimental Methods Specialty Conference (HMEM)*, Estes Park, Colorado, USA, July 28-August 1, pp. 1-7, 2002.

[17] HAJELA, P., and BERKE, L. "Neurobiological Computational Models in Structural Analysis and Design," *Computers and Structures*, vol. 41, pp. 657-667, 1991.

[18] HALFAWY, M., and FROESE, T. "Leveraging information technologies applications in the Canadian AEC/FM industry," *Presented at the Conference of Canadian Society for Civil Engineers*, May 30-Jun 2, Victoria, Canada, 2001.

[19] HARAN, J., DILLENBURG, J., and NELSON, P. "Realtime image processing algorithms for the detection of road and environmental conditions," *Presented at the*

Ninth International Conference on Applications of Advanced Technology in Transportation (AATT), Chicago, Illinois, USA, August 13-16, pp. 55-60, 2006.

[20] HIGGINS, C., and TURAN, O. "Imaging tools for evaluation of gusset plate connections in steel truss bridges," *Journal of Bridge Engineering*, vol. 18, pp. 380-387, 2013.

[21] HRYCIW, R., and JUNG, Y. "Accounting for void ratio variation in determination of grain size distribution by soil column image processing," *Presented at the GeoCongress 2008*, New Orleans, Louisiana, USA, March 9-12, pp. 966-973, 2008.

[22] JAZEBI, F., and RASHIDI, A. "An automated procedure for selecting project manager in construction firms," *Journal of Civil Engineering and Management*, vol. 19, pp. 97-106, 2013.

[23] JENG, D.S., BATENI, S.M., and LOCKETT, E. "Neural network assessment for scour depth around bridge piers," *Research Report No R855, Department of Civil Engineering, Environmental Fluids/Wind Group,* The University of Sydney, 2005.

[24] JENG, D.S. CHA, D.H., and BLUMENSTEIN, M. "Application of artificial neural network in civil engineering problems," *Presented at the International Internet-Processing-Systems-Interdisciplinaries (IPSI-2003) Conference*, October 5-11, Seveti-Stefan, Montenegro, 2003.

[25] KANG, D., LEE, J., CHOO, J., and YUN, T. "Pore directivity of soils subjected to shearing: numerical simulation and image processing," *Presented at the GeoCongress 2012*, Oakland, California, USA, March 25-29, pp. 2342-2351, 2012.

[26] KEMENY, J., DEVGAN, A., HAGAMAN, R., and WU, X. "Analysis of rock fragmentation using digital image processing," *Journal of Geotechnical Engineering*, vol. 119, pp. 1144-1160, 1993.

[27] LEE, S. "Automated defect recognition method by using digital image processing," *Presented at the 46th Annual International Conference by Associated Schools of Construction (ASC),* April 7-10, Boston, Massachusetts, USA, 2010.

[28] Lester woods general builders, "Bricklaying," www.lesterwoodsgeneralbuilders.co.uk.

[29] LIAO, R., and CHEN, Y. "The warning of rear-end accident on the highway based on image processing," *Presented at the 11th International Conference of Chinese Transportation Professionals (ICCTP)*, Nanjing, China, August 14-17,pp. 147-158, 2011.

[30] MAERZ, N.H. "Technical and computational aspects of the measurement of aggregate shape by digital image analysis," *Journal of Computing in Civil Engineering*, vol. 18, pp. 10-18, 2004.

[31] MALLIKARJUNA, C., PHANINDRA, A., and RAO, K. "Traffic data collection under mixed traffic conditions using video image processing," *Journal of Transportation Engineering*, vol. 135, pp. 174-182, 2009.

[32] MOGUERZA, J.M., and MUNOZ, A. "Support vector machines with applications," *Statistical Science*, vol. 21, pp. 322-336, 2006.

[33] MUKHERJEE, A., DESHPANDE, J., and ANMADA, J. "Prediction of buckling load of columns using artificial neural networks," *Journal of Structural Engineering*, vol.122, pp. 1385-1387, 1996.

[34] PUNMIA, B.C., JAIN, ASHOK K., and JAIN, ARUN K. "Mechanics of Materials," Laxmi Publications Pvt Limited, 2003.

[35] RASHIDI, A., JAZEBI, F., and BRILAKIS, I. "Neuro-fuzzy genetic system for selection of construction project managers," *Journal of Construction Engineering and Management*, vol. 137, pp. 17-29, 2011.

[36] REJAIE, A., and SHINOZUKA, M. "Reconnaissance of Golcuk 1999 earthquake damage using satellite images," *Journal of Aerospace Engineering*, vol. 17, pp. 20-25, 2004.

[37] RIVARD, H. "A survey on the impact of information technology in the Canadian architecture, engineering and construction industry," *Electronic Journal of Information Technology in Construction*, vol. 5, pp. 37-56, 2000.

[38] SANAD, A., and SAKA, M. "Prediction of ultimate shear strength of reinforced concrete deep beams using neural networks," *Journal of Structural Engineering*, vol. 127, pp. 818-828, 2001.

[39] SHAHIN, M.A., JAKSA, M.B., and MAIER, H.R. "Artificial neural network applications in geotechnical engineering," *Australian Geomechanics*, vol. 36, pp. 49-62, 2001.

[40] SINHA, S.K. "Automated underground pipe inspection using a unified image processing and artificial intelligence methodology," *A thesis presented to the University of Waterloo in fulfillment of the thesis requirement*, Ontario, Canada, 2000.

[41] SON, H., KIM, C., and KIM, C. "Automated color model–based concrete detection in construction-site images by using machine learning algorithms," *Journal of Computing in Civil Engineering*,vol. 26, pp. 421-433, 2012.

[42] TANG, J., and SUN, B. "An integrated digital image processing pavement management information system," *Presented at the International Conference of Logistics Engineering and Management (ICLEM 2012)*, Chengdu, China, October 8-10, pp. 1300-1306, 2012.

[43] WIGAN, M. "Image-Processing Techniques Applied to Road Problems," *Journal of Transportation Engineering*, vol. 118, pp. 62-83, 1992.

[44] WILMOT, C., and MEI, B. "Neural Network Modeling of Highway Construction Costs," *Journal of Construction Engineering and Management*, vol. 131, pp. 765-771, 2005.

[45] WU, G., HIGUCHI, K., and MERONEY, R. "Application of digital image processing in wind engineering," *Presented at the 8th International Conference on wind engineering*, July 8-12, Ontario, Canada, pp. 1-12, 1991.

[46] YOSHIDA, J., ABE, M., FUJINO, Y., and SUJEEWA, L. "Measurement method for continua by image processing," *Journal of Structural Engineering*, vol. 130, pp. 1145-1156, 2004.

[47] ZHAO, L., and NIU, L. "Study on key techniques of image processing and automatic recognition of tunnel cracks," *Presented at the International Conference of Logistics Engineering and Management (ICLEM 2012)*, Chengdu, China, October 8-10, pp. 427-433, 2012.

[48] ZHU, Z., and BRILAKIS, I. "Parameter optimization for automated concrete detection in image data," *Automation in Construction*, vol. 19, pp. 944-953, 2010.

[49] ZHU, Z., and BRILAKIS, I. “Concrete column recognition in images and videos,” *Journal of Computing in Civil Engineering*, vol. 24, pp. 478-487, 2010.